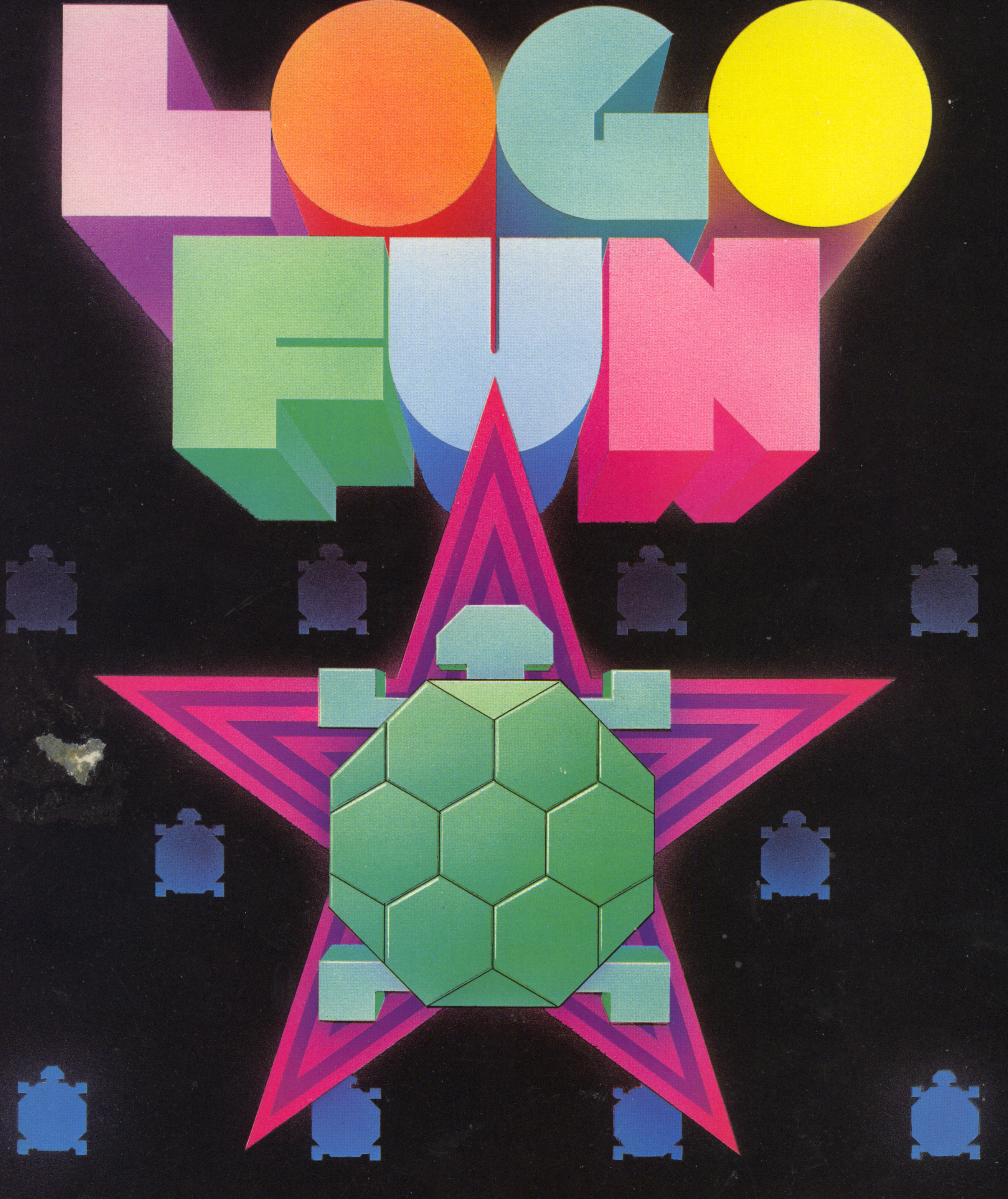
Scholastic 0-590-33243-0 / $5.95

k POWER

Fun and easy LOGO programs
for your Apple, Atari, and TI computers.

Pat Parker and Teresa Kennedy

LOGO FUN



k POWER

Logo Fun

Pat Parker
and
Teresa Kennedy

SCHOLASTIC INC.
New York Toronto London Auckland Sydney Tokyo

No part of this publication may be reproduced in whole or in part, or stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission of the publisher. For information regarding permission, write to Scholastic Inc., 730 Broadway, New York, NY 10003.

ISBN 0-590-33243-0

Copyright © 1985 by Pat Parker and Teresa Kennedy.
All rights reserved. Published by Scholastic Inc.

Designed by Gene Siegel

12 11 10 9 8 7 6 5 4 3 2 1 1 5 6 7 8 9/8 0/9
14

Printed in the U.S.A.

Logo Fun

Computer Craziness by Paul Somerson and Stephen Manes

Computer Monsters by Stephen Manes and Paul Somerson

Computer Olympics by Stephen Manes and Paul Somerson

Computer Space Adventures by Paul Somerson and Stephen Manes

Logo Fun by Pat Parker and Teresa Kennedy

Contents

INTRODUCTION 1

Note to Parents and Teachers; Note to Kids

CHAPTER ONE: TURTLE TALK 5

Logo Primitives and How They Work

CHAPTER TWO: TEACHING THE TURTLE 31

Writing a Procedure; Nesting Procedures; Saving Your Work

CHAPTER THREE: ALL ABOUT COLOR 51

Using Logo with Color Monitors; Logo Color Commands

CHAPTER FOUR: AROUND AND AROUND AND OVER AND OVER 67

Circles and Arcs; Variables and Recursion

CHAPTER FIVE: TURTLES AND MORE TURTLES 87

Other Ways to Move the Turtle; Dynamic Turtles; Multiple Turtles; Shapes and Sprites

CHAPTER SIX: GAMES AND CHALLENGES 101

Introduction

NOTE TO PARENTS AND TEACHERS:

WELCOME TO LOGO! Logo is a unique computer programming language first created in the late 1960s. Recently it has become available for microcomputers, and in the short period of time that it's been used in homes and classrooms, it has created a real revolution in computer education. In fact, many experts believe that Logo will replace BASIC as the introductory programming language. Logo is amazingly simple to use, yet it is a powerful programming tool. A few simple commands can result in animated pictures, explosions of light and color—even your own video games! It doesn't require sophisticated programming skills, but by mastering some simple concepts and commands Logo can, and does, result in some very sophisticated programs.

Logo is both easy to learn and easy to teach. The important thing to remember about working with Logo is that it is as much about *how* people learn as *what* people learn. Different people learn in different ways: some of us are methodical, some intuitive; some are independent, others in need of guidance. There is information for every kind of learner in this book. For those who prefer to learn by example, we have included some ready-made programs. For those who prefer to experiment, we have tried to give advice wherever we could.

We want to stress that there is no right or wrong way to learn Logo. There are no mistakes or failures—rather, if you have a problem with a program, you and the student can work together to find the solution. We want this book to be an aid in learning Logo, not a book of preprogrammed exercises to be memorized and repeated like multiplication tables. It isn't necessary that it be studied from cover to cover before attempting a program. We would hope instead that each section be covered as the individual is ready to learn a particular concept. Some students, for example, will want to learn about recursion before they learn about programming color; others will not. The point is that every Logo student, adult or child, is encouraged to go through the book and look for what he or she needs to know to complete a particular project or solve a problem.

We have included Logo commands and programs for four different versions of Logo software: **Apple Logo** and **Terrapin/Krell Logo** for Apple II computers, **Atari Logo** for Atari computers, and **TI Logo** for Texas Instruments computers. Specific commands for each of these versions have been included directly in the text or in the form of footnotes. In working with this book, look for and use only those commands for your version of Logo. The variations in Logo software are considerable, so it might be a good idea to at least familiarize yourself and your students with the individual manuals that are included with your version of Logo. Doing so will give you a head start in grasping these programs.

Logo is a language that is flexible, versatile, and fun. With the aid of this book, you and your students can create complex and exciting visual images, games, and experiments. And in doing so, the computer user is introduced to equally exciting learning concepts. For in a way, learning about Logo is learning about everything—how things move in space, physical laws and principles, and basic problem solving. Creating a picture or a video game can result in a solid grasp of mathematics or geometry, or serve as an introduction to Einstein's theory of relativity. But in any case, learning with Logo should result in a sense of achievement—that special satisfaction and pleasure that comes when one is involved in the process of finding out about the world. And perhaps that is the greatest learning concept of all.

NOTE TO KIDS:

You're about to meet a new kind of turtle. Think of this turtle as a creature that lives inside your computer. The Turtle will come out to help you create pictures and programs whenever you tell it to. The Turtle can be a great helper, but it isn't very smart. It can only do what you tell it to do—nothing more, nothing less. The way to tell the Turtle what you want it to do is by learning to speak its language. That language is called Logo.

As you're just beginning to learn Logo, take the time to do some experimenting. Try out some different commands to see what they mean and what they can make the Turtle do for you before writing a complete program. Draw some lines to see how the Turtle moves. If the Turtle needs to know more before it can complete a command, it will ask you for more information. If you don't understand how to get the Turtle to do what you want it to do, ask for help. There's always a way—you just have to find it.

We have included commands for some different versions of Logo here, but if you don't find your version, check your program manual. If you want to try one of our programs, look for the right commands to use. We have included Logo commands and programs for four different versions of Logo software: **Apple Logo** and **Terrapin/Krell Logo** for Apple II computers, **Atari Logo** for Atari computers, and **TI Logo** for Texas Instruments computers. All of the programs in this book will work, but some of the commands to make them work will be different for different versions of Logo.

Use this book like you might use an encyclopedia or a dictionary. If you're writing a program and you don't know how to make the Turtle do what you want it to do, look it up. Flip back and forth, or check the index to find what you need. Most of the programs here were written by kids just like you. You can learn Logo by using these programs or by writing programs of your own. The important thing to remember is, enjoy yourself!

1

Turtle Talk

TURNING LOGO ON

To meet the Turtle, you must follow a few simple steps. Depending on what kind of machine you're working with, your Logo software will either be on a floppy disk or a cartridge. Carefully insert the software into your computer or disk drive and then turn on the computer and the monitor. It will take a few seconds before you are able to begin.

When Logo is ready, a welcome message will appear in one corner of your screen. Under the message, there will be a question mark and a small rectangle or line called a cursor. This means that Logo is ready for your first command.

MEET THE TURTLE

To get the Turtle to appear on your screen, you will need to enter the correct command for your machine from the list below:

APPLE LOGO: SHOWTURTLE or the abbreviation ST

ATARI LOGO: ST

TEXAS INSTRUMENTS (TI) LOGO: TELL TURTLE

TERRAPIN/KRELL LOGO (T/K): DRAW

Then press either the key marked RETURN or ENTER. This tells the Turtle that it's time to do what you've told it to do. This is called *executing a com-*

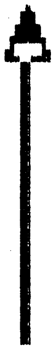
mand. From now on, every time you type in a command, be sure to press the RETURN or ENTER key to get the Turtle to execute your commands.

COMMANDS

The Turtle moves around on your computer screen according to the *commands* you give it. These commands tell the Turtle how far to move and in which direction. The Turtle understands some commands already. It has been taught to carry out these commands by the computer programmers who invented Logo. In a very short time, you will be a computer programmer too, and will be teaching the Turtle new commands. Try some experiments with the commands the Turtle already knows. These are called *primitives.* A list of primitives follows, with a description of what each primitive does. Read on and see how easy it is to "talk" to the Turtle!

FORWARD (FD)

To get the Turtle to move forward, type the word FORWARD or the abbreviation FD followed by a space and a number. The number tells the Turtle how far forward to go. The Turtle travels in turtle steps, and the number you type in after your command tells it how many turtle steps to take.



FD 65

Try typing in the FORWARD command followed by some different numbers. Remember to leave a space between the command and the number, and to press RETURN or ENTER each time you want the Turtle to move.

You may have noticed that fairly large numbers send the Turtle off the screen, around, and back on again from the other side. This is called *wrapping around.* If your Turtle disappears, keep telling it to go FORWARD until it reappears. If your number is really big, the Turtle will not be able to go that far because there will not be enough room in the computer for that many turtle steps.

If that happens, an error message will appear at the bottom of your screen. In this case, the error message will probably read, "TURTLE OUT OF BOUNDS," or "OUT OF INK." For more on error messages, see page 26.

CLEARSCREEN (CS)

By now there are probably lines on your screen. To make them go away, type the command CLEARSCREEN or the abbreviation CS. In most versions, CS erases all lines on your screen and sends the Turtle back to its original position in the middle of your screen. In **TK Logo**, CLEARSCREEN will erase all lines on your screen; to send the Turtle back to the middle of your screen, you must type DRAW.

RIGHT (RT) and LEFT (LT)

The Turtle can also turn. The turning commands are RIGHT or RT for right turns, and LEFT or LT for left turns. These commands are followed by a space and number. When you type in the number after a right or left command, you are telling the Turtle how many degrees you want it to turn. For example, RT 75 turns the Turtle to the right 75 degrees. If you turn the Turtle RT or LT, then enter a FORWARD command followed by a number, the Turtle will move forward in a new direction. For example:

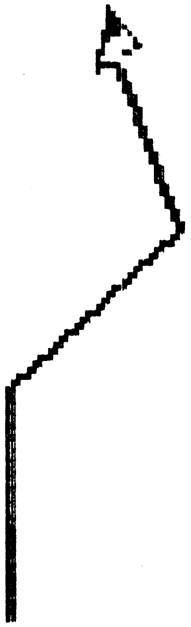
```
RT 75 FD 55.
```

BACK (BK)

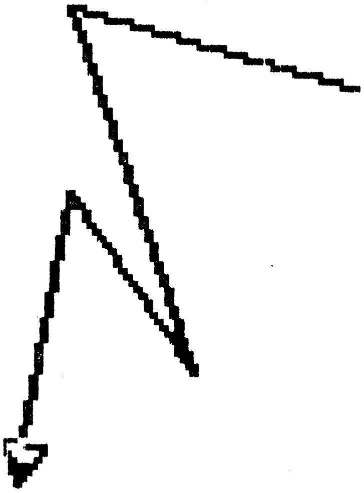
The Turtle goes backwards, too, using the command BACK or BK followed by a space and a number. Experiment with some combinations of commands—BK 25 RT 90 FD 50. (Remember: If you want to clear your screen, type CS.)

BK 65

Using the commands you now know, can you get the Turtle to draw a picture for you? You can type several commands on one line if you want. Just remember to leave a space between each number and each command.



```
FD 45 RT 45 FD 55 LT 45 FD 35
```



```
LT 75 FD 56 RT 56 BK 75 LT 16 FD 50 LT 135 FD 65
```

Can you figure out how to draw a square with the Turtle? How about a triangle?

HOME

The center of the screen is called HOME. There is no abbreviation or short form of this command. It is different from the CLEARSCREEN or CLEAN commands because it does not erase all lines from the screen, and in some versions draws a line from its position back to the center.

CLEAN

CLEAN is a command found in both **Apple** and **Atari** versions of Logo which cleans all lines off your screen without moving the Turtle from its position on the screen. There is no short form or abbreviation of this command.

PENUP (PU) and PENDOWN (PD)

So far the Turtle has been drawing lines on your screen. But it can also move without drawing any lines. To make the Turtle move without drawing lines, use the command PENUP or PU. But it's important to remember that in order for the Turtle to draw again after using the PENUP command, you must use the command PENDOWN or PD. Try some experiments alternating the PU and PD commands.

PENERASE

PENERASE is a command that goes back over a line you've just drawn and erases it. Suppose you've just typed FD 50 and pressed RETURN and decided that you don't like the way that line looks. You would erase it by typing PENERASE or PE BACK 50. Or if you only want to erase half that line, type PE BK 25 and half the line will disappear. PE picks up the Turtle's pen like PU, so always remember to type PD or PENDOWN when you are finished using the PENERASE command.

REPEAT

The REPEAT command gives you a lot of creative power when drawing and making designs. Here's why: Suppose you wanted to draw a simple square, 40 turtle steps long on each side. You could type:

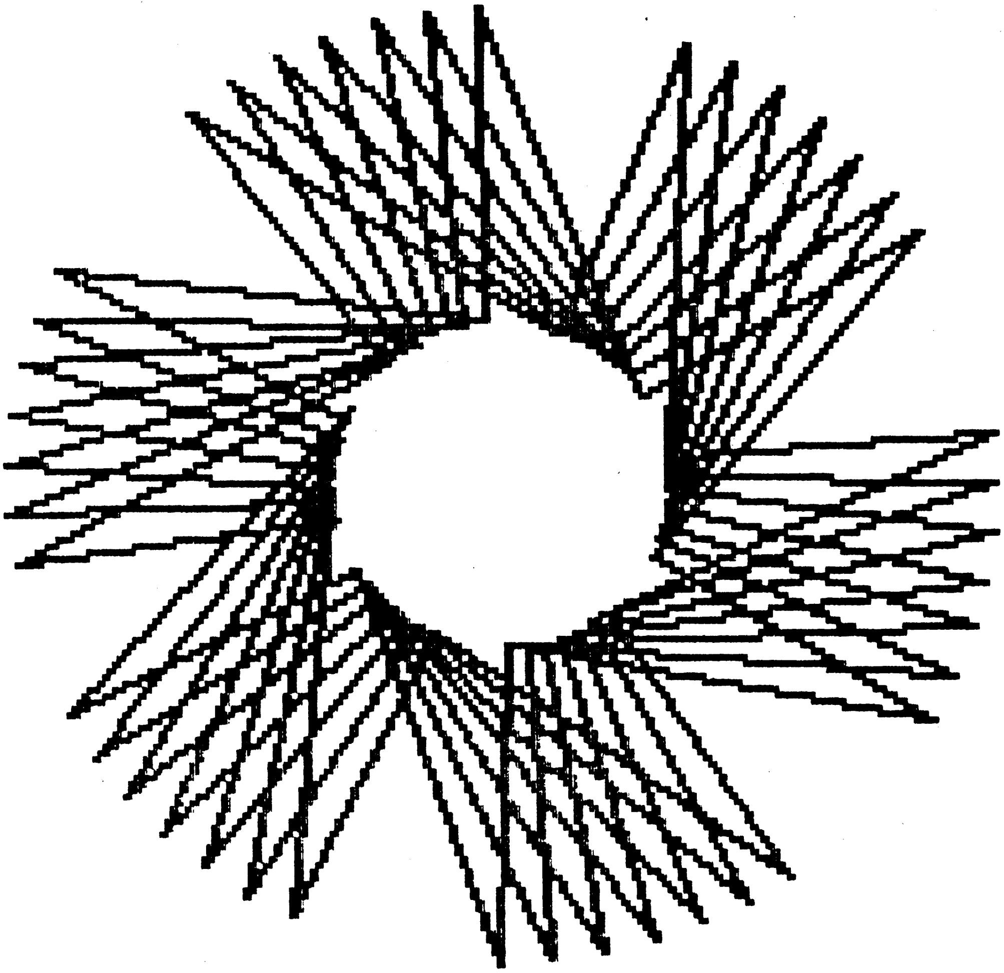
```
FD 40 RT 90
FD 40 RT 90
FD 40 RT 90
FD 40 RT 90
```

Or instead you could use the REPEAT command and type REPEAT 4 [FD 40 RT 90]. This tells the Turtle to make each side of the square with one simple command. REPEAT is especially useful when you want to make up complicated games and designs, because it does away with a lot of typing. To use the REPEAT command, first type REPEAT, then a space, and then a number; then list the things you want REPEATed inside brackets [ ]:

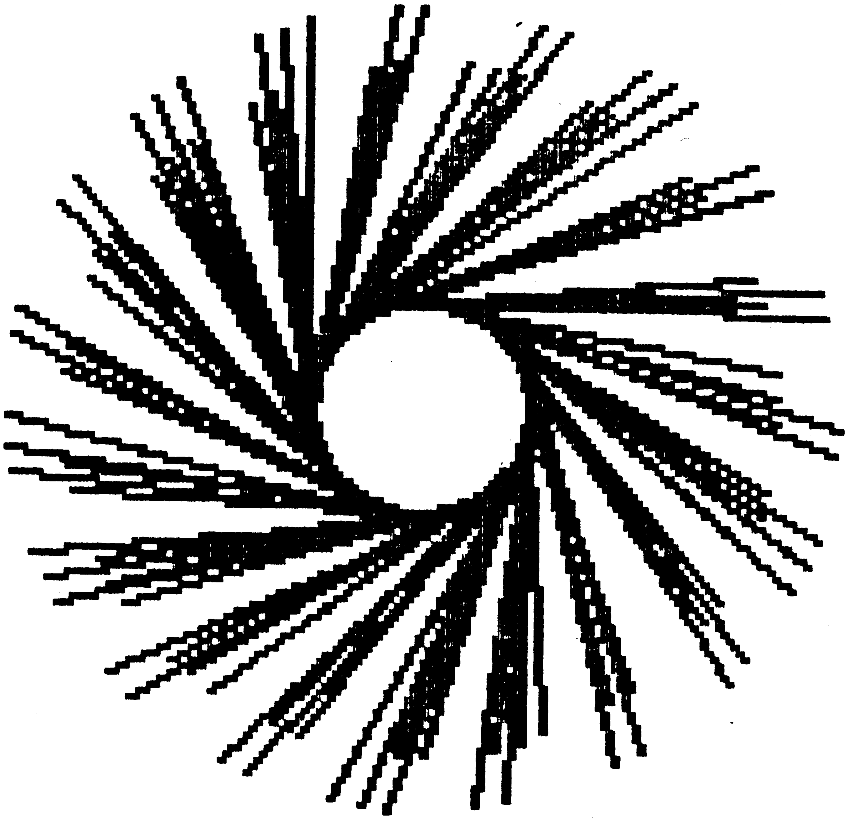
> **APPLE II and TI Plus:** Press the SHIFT key, hold it down, and press N for the left bracket [and M for the right bracket].
>
> **APPLE IIe:** Bracket keys are marked.
>
> **ATARI:** Press the SHIFT key, hold it down and press the bracket keys marked [and].
>
> **TI:** Press the FCTN key, hold it down, and press R for the left bracket and T for the right bracket.

The fun of REPEAT begins when you start changing the numbers and your lists of commands. You can put any list of commands inside the brackets and repeat them any number of times. Get familiar with the REPEAT command, and before long, your designs will come out exactly as you picture them. For now, though, it's a good idea to try some experiments and get used to how the REPEAT command works. If the Turtle can't carry out your commands for some reason, it will tell you with an error message. (See page 26.) You can always change your commands so the Turtle will understand.

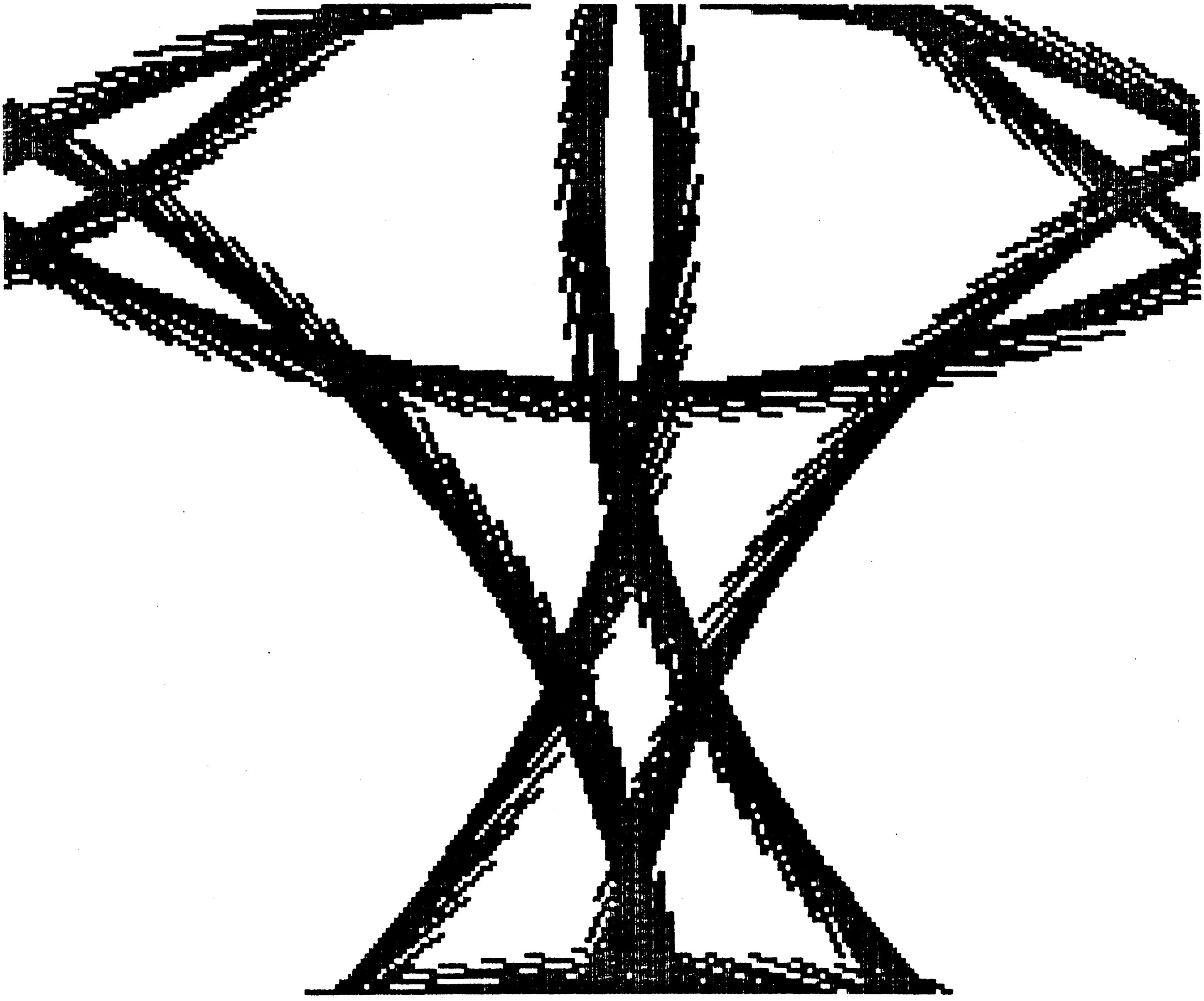
The programs that follow were written by kids using the REPEAT command. Some of these designs were discovered by accident during their first day of learning Logo! Experiment with these designs and some of your own, and see what the REPEAT command can do.



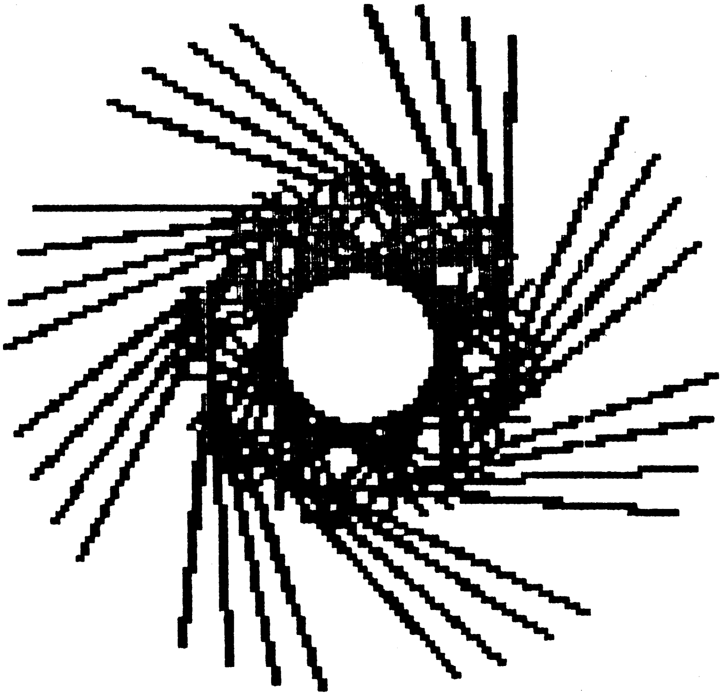
REPEAT 42 [RT 82 FD 63 LT 21 BK 83]



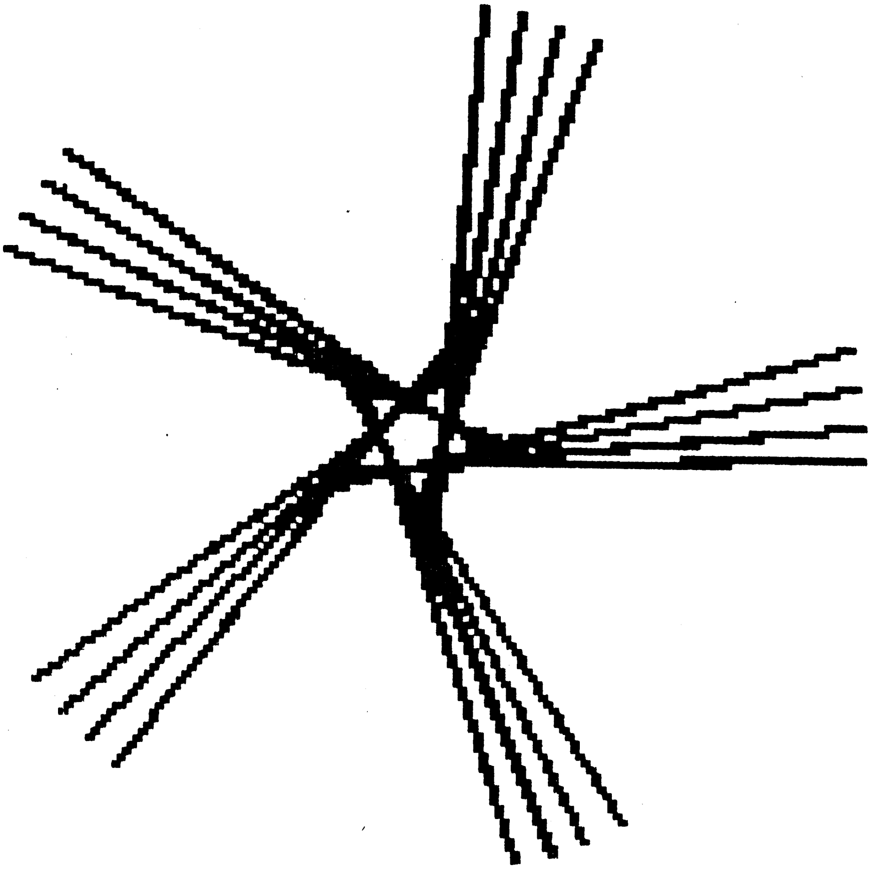
REPEAT 45 [RT 53 BK 78 FD 59 RT 23 BK 79 FD 68]



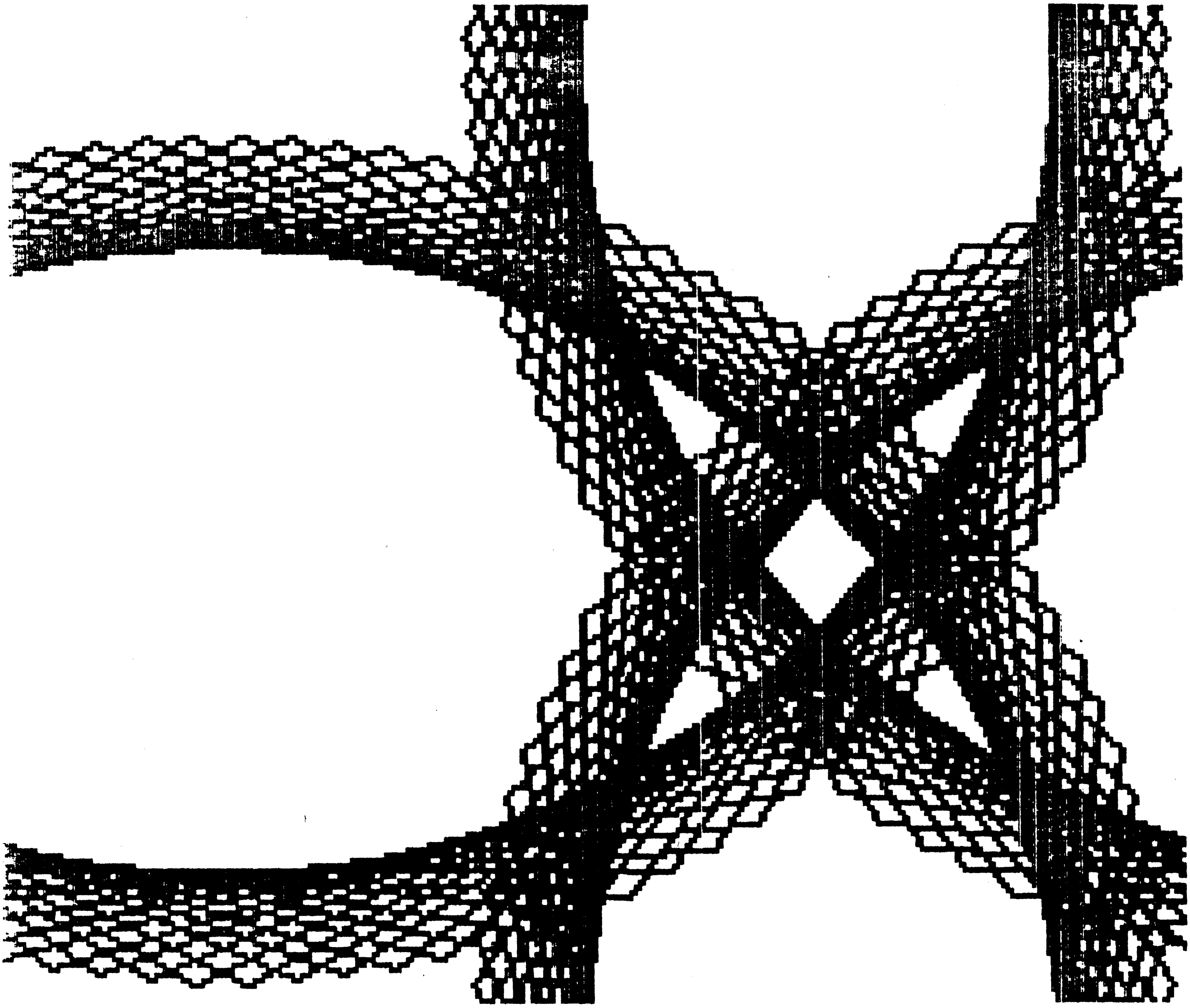
REPEAT 200 [FD 80 RT 2 BK 70]



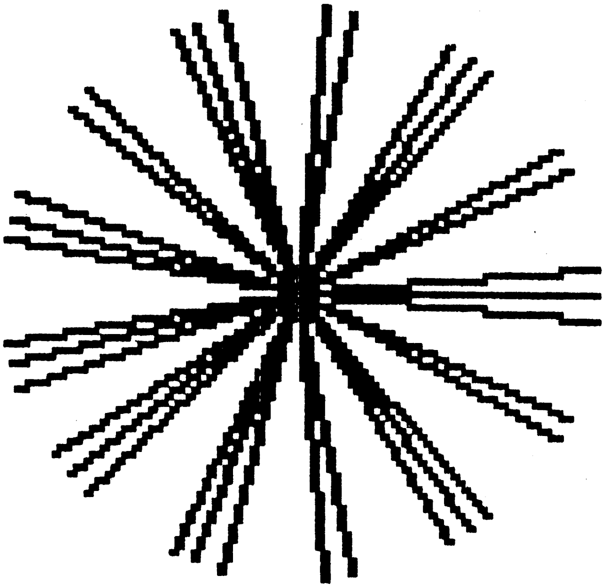
REPEAT 32 [FD 63 BK 56 RT 52 FD 65 BK 43 LT 98]



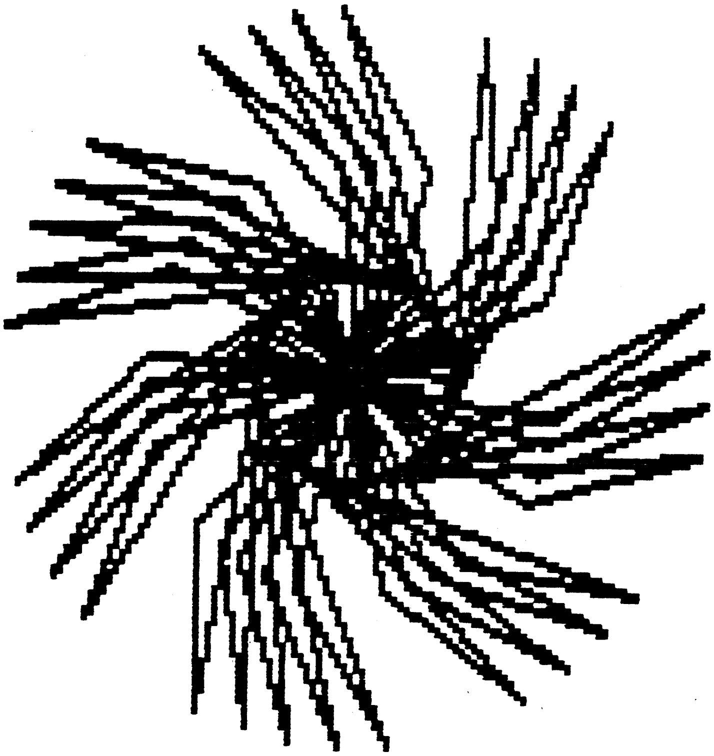
REPEAT 20 [RT 217 FD 100 BK 67]



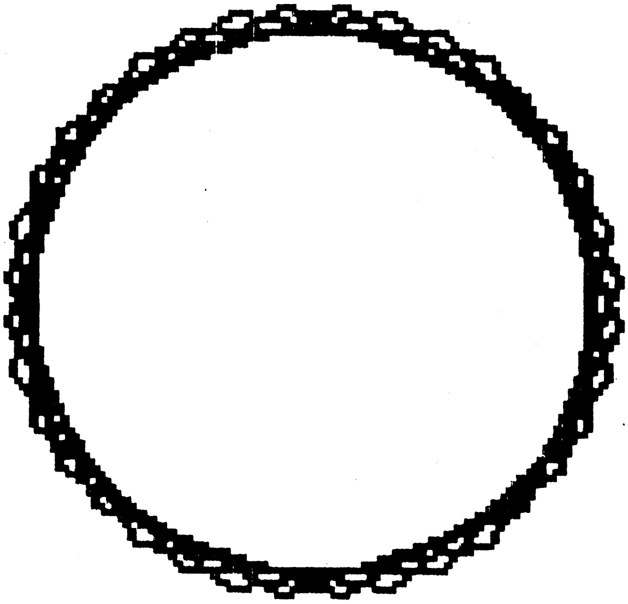
REPEAT 145 [BK 212 LT 303]



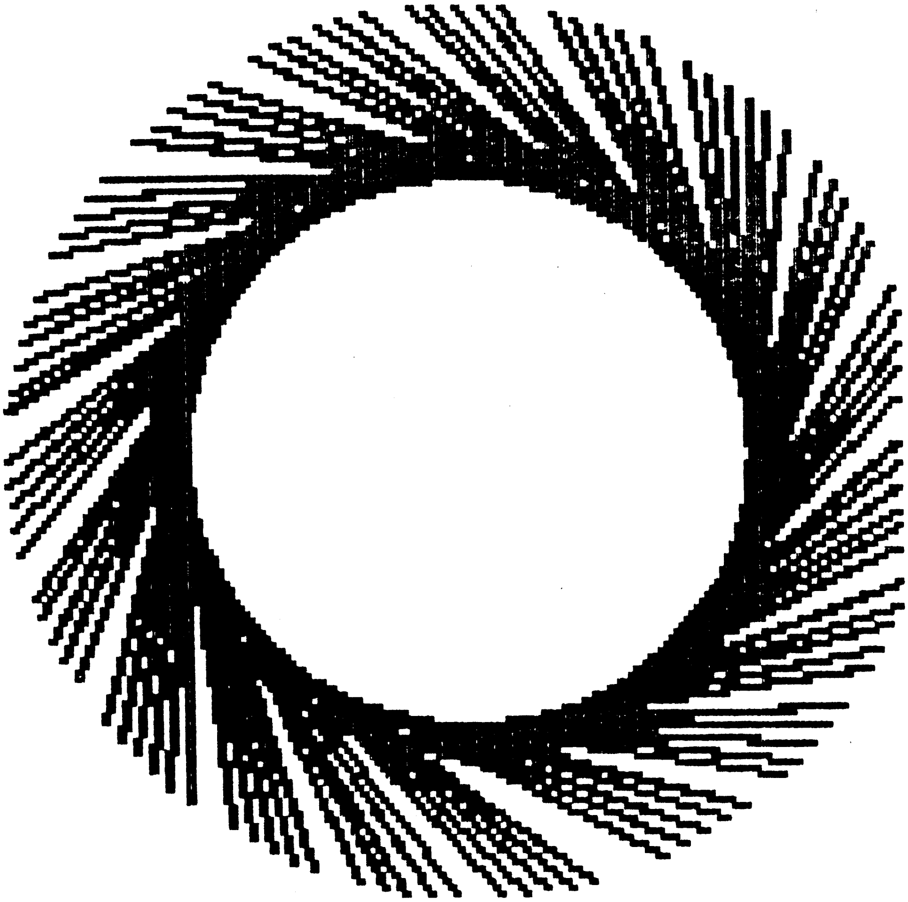
REPEAT 34 [RT 55 FD 55 BK 55]



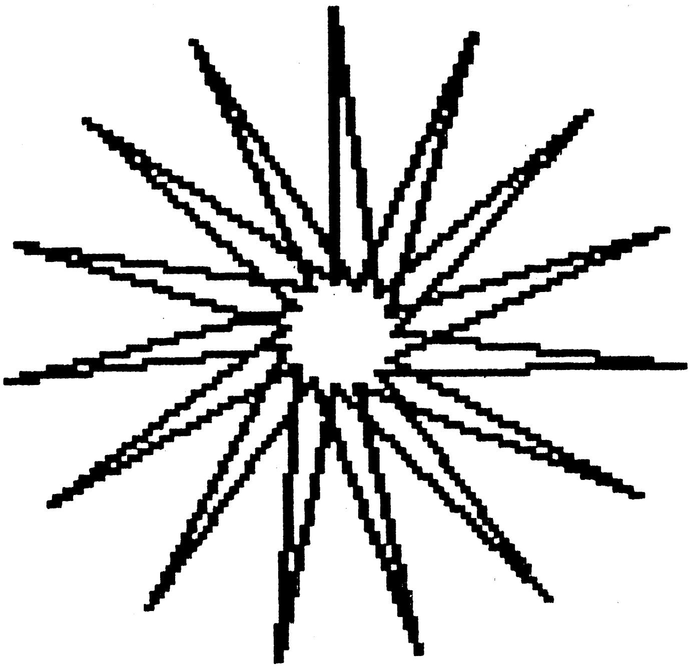
REPEAT 30 [FD 34 RT 68 BK 74 LT 10 FD 36 RT 46 FD 47 BK 19]



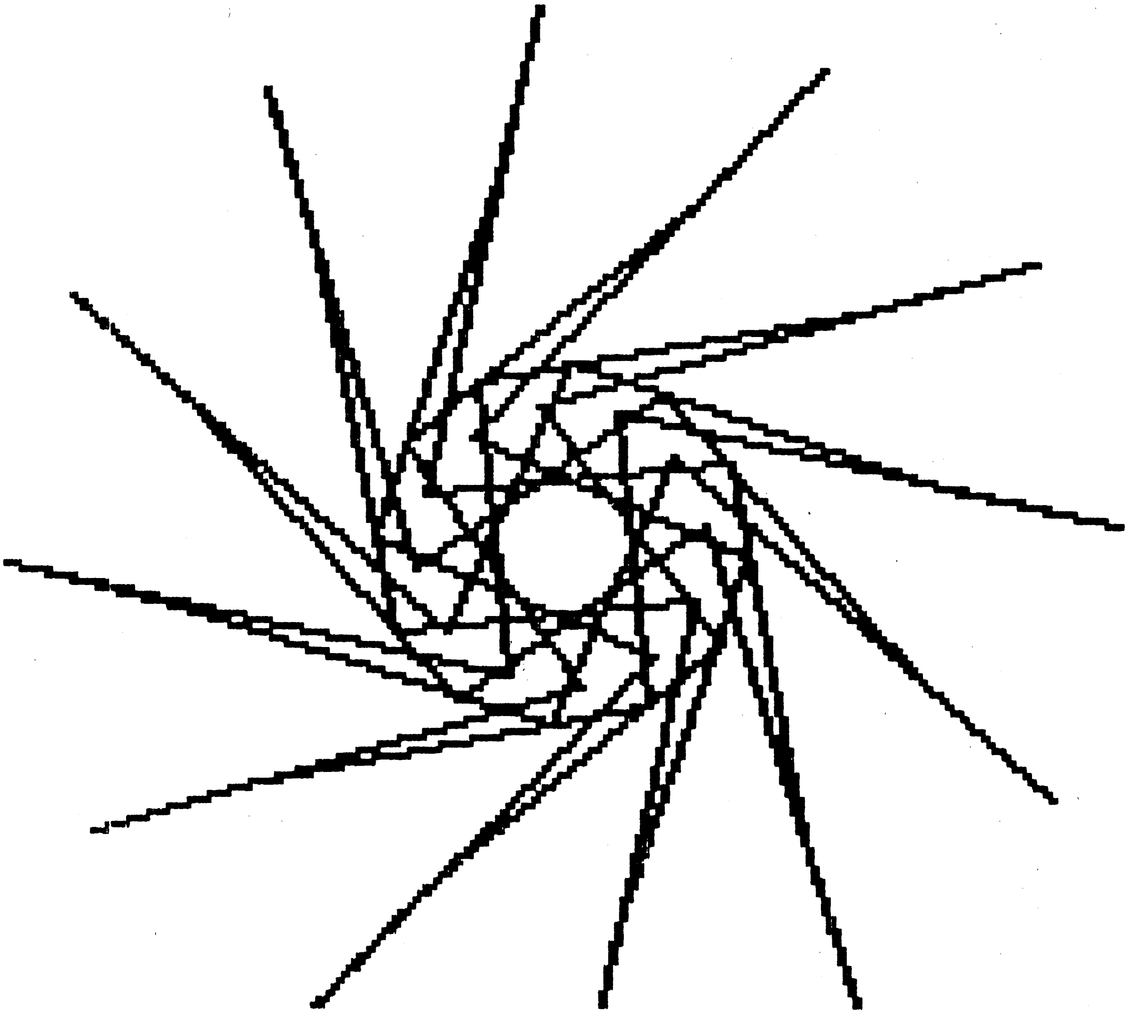
REPEAT 50 [FD 50 RT 50]



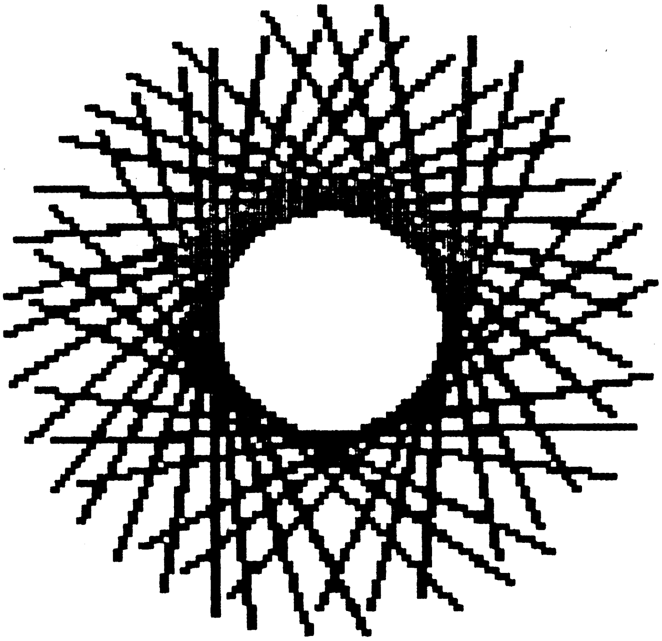
REPEAT 100 [BK 40 FD 100 RT 57]



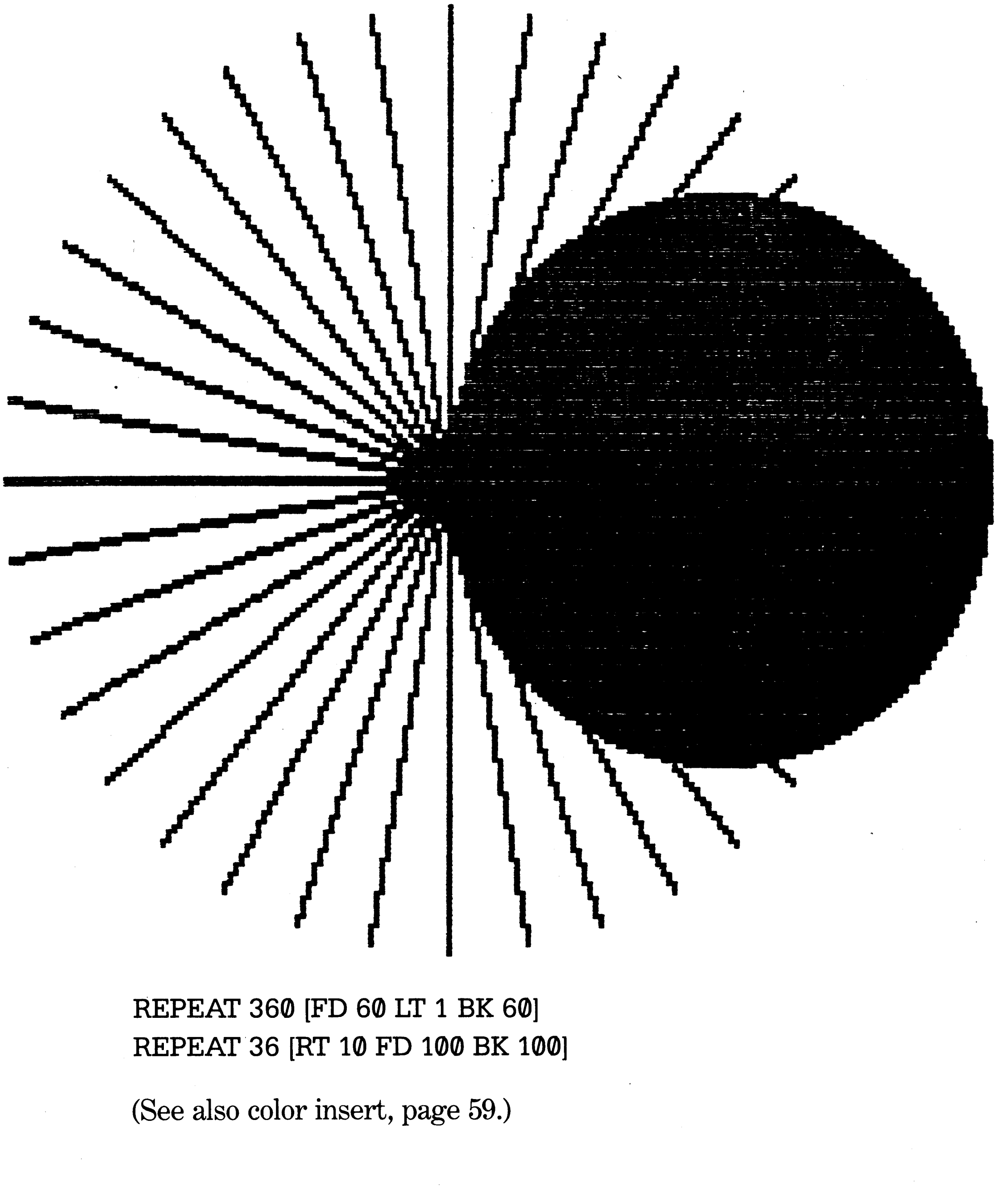
REPEAT 15 [FD 53 LT 56 BK 56 RT 8]



REPEAT 12 [LT 36 FD 59 LT 95 BK 76 LT 6 BK 35 FD 100 LT 73]

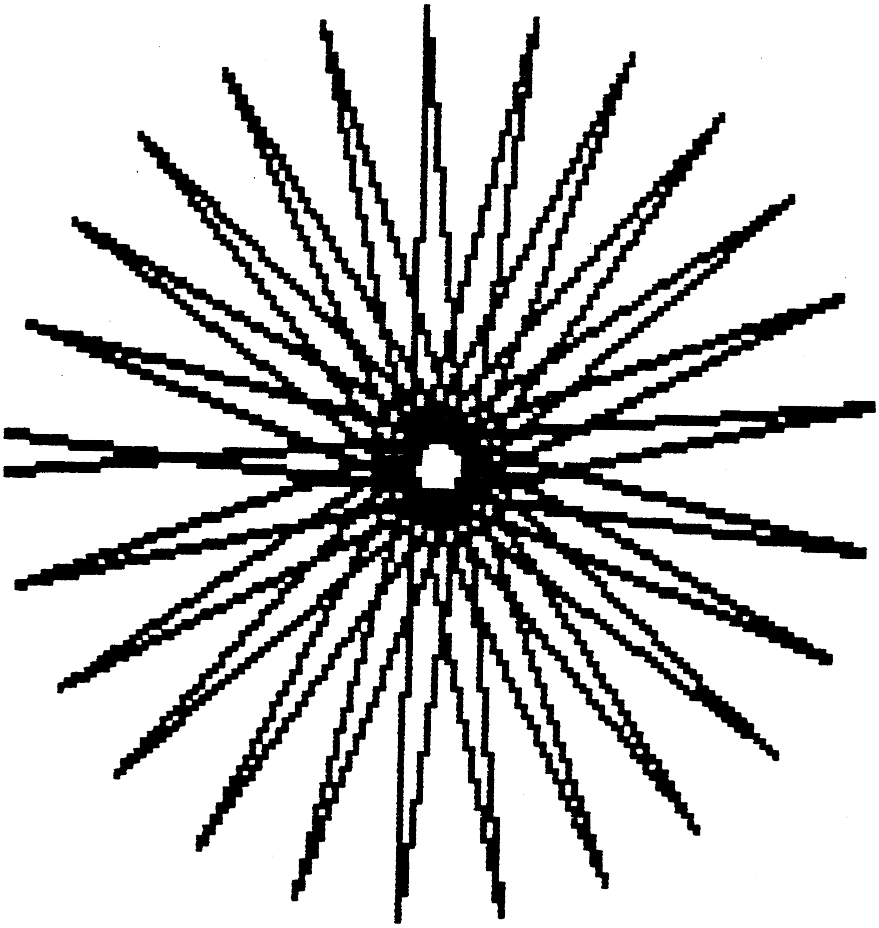


REPEAT 33 [FD 51 BK 112 FD 46 RT 37]



```
REPEAT 360 [FD 60 LT 1 BK 60]
REPEAT 36 [RT 10 FD 100 BK 100]
```

(See also color insert, page 59.)



REPEAT 25 [RT 187 FD 168]

HIDETURTLE (HT) and SHOWTURTLE (ST)

The command HIDETURTLE (HT) makes the Turtle disappear. It will still draw and move but it will be invisible. In some versions of Logo, HT allows the designs to be drawn faster than they would if the Turtle were showing on the screen. SHOWTURTLE (ST) is the command to use when you want the Turtle to reappear on your screen.

FULLSCREEN

APPLE and T/K LOGO: FULLSCREEN

ATARI LOGO: FS

This command covers the text at the bottom of the screen, allowing your design to fill the entire screen.

SPLITSCREEN

APPLE and T/K LOGO: SPLITSCREEN

ATARI LOGO: SS

This command allows for five lines of text at the bottom of the screen, while the top of the screen is available for graphics.

TEXTSCREEN

APPLE and T/K LOGO: TEXTSCREEN

ATARI LOGO: TS

This command provides the whole screen for text. These commands are not available in **TI Logo.**

ERROR MESSAGES:

What happens if you make a typing mistake in Logo? Usually you will get an error message telling you that the Turtle doesn't know how to complete your command. Error messages are nothing to worry about. They are there to help you learn Logo, and learn about what the Turtle can and cannot do for you. If, for example, you typed FD122, without leaving a space between the command and the number, you will probably get a message like "I DON'T KNOW HOW TO FD122." The Turtle is fussy about typing. When you get an error message like this, it simply means that the Turtle is telling you to type your command in once again exactly right. Then it will be able to obey your commands.

What happens if you enter a command without a number—FD (RETURN), for example? Some versions of Logo will give the message "TELL ME MORE," while others will give the message, "NOT ENOUGH INPUTS TO FD." They both mean the same thing—that the Turtle needs a number before it can make a move. An *input* is any information that you enter into the computer.

Let's say you enter a string of commands like the one below. What will happen? Can you tell where the mistake is?

```
FD LT 22 BK 57
```

Most versions of Logo will say, "FD DIDN'T OUTPUT TO LT." This means that a number was left out after FD and the Turtle didn't know what to do with the FD command. An *output* is the information the computer gives you in response to your command. When you input FD 50 and press RETURN or ENTER, the output is a line 50 turtle steps long.

CRASHING

Each keyboard also has a key or keys which *crash* Logo. This means that these keys will *lose* Logo and lose all of your work for you. Therefore, BE CAREFUL not to press these keys unless you really mean it:

APPLE KEYBOARDS: RESET

ATARI KEYBOARDS: RESET

TI KEYBOARDS: FCTN and the key marked QUIT (above the=)

THE TOTAL TURTLE TRIP AND THE MAGIC 360

As we talked about before, the Turtle turns in degrees. A *degree* is a number used to measure the amount of a turn. For example, if you give the command RT 45, the Turtle will turn right 45 degrees.

If you look at some of the shapes we have worked with so far, you will notice something curious about them. No matter which shape you draw or how many sides it has, *the number of degrees in the turns the Turtle makes to get around any shape and back to where it started will add up to 360.* When the Turtle has completed 360 degrees in turns, we call it a Total Turtle Trip:

To make a square: REPEAT 4 [FD 50 RT 90]
There are four right turns, each 90 degrees; 4×90=360.
How about some other shapes?
To make a hexagon: REPEAT 6 [FD 50 RT 60]

A hexagon has six sides. Therefore, the Turtle has to turn six times to make the shape. The number of degrees in each turn for a hexagon is 60; 6×60=360.

To figure out the number of degrees in each turn for any shape you want to make, take the number of sides of the shape and divide it into 360. For example, an octagon has eight sides; 360÷8=45. So the amount of each turn would be 45 degrees.

Other Useful Numbers

180—This number is one-half of 360. So if you only want to go halfway around a shape, have the number of degrees in your turns add up to 180.

120—This number is one-third of 360. This is the amount to turn when making a shape with three equal sides—a triangle.

90—This number is one-fourth of 360. If you want to go only a quarter of the way around a shape, have the number of degrees in your turns add up to 90. A 90-degree angle is called a right angle. That means that a 90-degree turn creates an angle that looks like a corner.

45—This number is one-eighth of 360. This is the amount to turn when making a shape with eight equal sides.

PLAYING TURTLE

It is sometimes confusing trying to figure out which way to tell the Turtle to turn, or how to tell it to do just what you want to complete a drawing or design. This is where playing Turtle can come in very handy. All you have to do to play Turtle is to pretend that you are the Turtle. Stand up and walk through your design to see which way your body turns, or have a friend be the Turtle and walk through the design as you give him commands. When you think you have the answer, try giving the Turtle on your screen that command. Try it—it works!

WORDS TO REMEMBER: CHAPTER ONE

WELCOME MESSAGE—A message that appears in the upper left hand corner of your computer screen, letting you know that Logo is ready to go.

CURSOR—A small square or line, which moves across the screen as you type to help you keep track of where you are.

COMMAND—An instruction from you to the Turtle that lets it know what you want it to do.

PRIMITIVE—A built-in Logo command that the Turtle already understands.

INPUT—The numbers and words you enter into the computer.

OUTPUT—The information that the computer gives you as a response to your commands.

WRAPPING AROUND—The movement of the Turtle off an edge of the screen and back on again from the other side.

ERROR MESSAGE—A message from Logo letting you know that it doesn't understand the information you just entered.

DEGREE—The measurement of the amount of an angle or turn.

CRASH—To lose Logo by hitting a key or keys. If you crash Logo accidently, it loses all your work and you must start over.

COMMANDS TO REMEMBER: CHAPTER ONE

PRIMITIVE	ABBREVIATION	INPUTS NEEDED	FUNCTION	VERSION
FORWARD	FD	NUMBER	MOVES TURTLE FORWARD IN TURTLE STEPS	ALL
BACK	BK	NUMBER	MOVES TURTLE BACKWARD IN TURTLE STEPS	ALL
RIGHT	RT	NUMBER	TURNS TURTLE TO THE RIGHT BY DEGREES	ALL
LEFT	LT	NUMBER	TURNS TURTLE TO THE LEFT BY DEGREES	ALL
REPEAT	NONE	NUMBER AND LIST OF INSTRUCTIONS	REPEATS LIST OF INSTRUCTIONS	ALL
PENDOWN	PD	NONE	TURTLE DRAWS AS IT MOVES	ALL
PENUP	PU	NONE	TURTLE MOVES WITHOUT DRAWING	ALL
HIDETURTLE	HT	NONE	MAKES TURTLE INVISIBLE	ALL
SHOWTURTLE	ST	NONE	TURTLE IS VISIBLE	ALL
CLEARSCREEN	CS	NONE	CLEARS LINES FROM SCREEN AND SENDS TURTLE HOME	ALL
HOME	NONE	NONE	SENDS TURTLE TO CENTER OF SCREEN	ALL
DRAW	NONE	NONE	CLEARS SCREEN AND SENDS TURTLE HOME	TERRAPIN/ KRELL
CLEAN	NONE	NONE	CLEARS LINES FROM SCREEN WITHOUT MOVING TURTLE	APPLE ATARI
TELL TURTLE	NONE	NONE	PLACES TURTLE IN SCREEN CENTER	TEXAS INSTRUMENTS

PRIMITIVE	ABBREVIATION	INPUTS NEEDED	FUNCTION	VERSION
PENERASE	PE	NONE	TURTLE'S PEN BECOMES AN ERASER	ALL
FULLSCREEN	FS (Atari only)	NONE	ALLOWS WHOLE SCREEN FOR GRAPHICS	APPLE, ATARI T/K
SPLITSCREEN	SS (Atari only)	NONE	SPLITS SCREEN FOR TEXT AND GRAPHICS	APPLE, ATARI T/K
TEXTSCREEN	TS (Atari only)	NONE	ALLOWS WHOLE SCREEN FOR TEXT	APPLE, ATARI T/K

2

Teaching the Turtle

Now that you know how to use some of the commands the Turtle already knows, you can teach it new commands for drawing and making designs. Teaching the Turtle is easy—you just have to know how.

Before the Turtle can use a new command, it has to know the definition of that command. The way to teach the Turtle a new command is by writing a *procedure.*

WRITING A PROCEDURE

A *procedure* is a computer program. It tells the Turtle, step by step, exactly what you want it to do when you give it your new command.

A procedure has three parts:

1. A name
2. A series of instructions
3. An END

Say, for example, that you want the Turtle to draw a square. If you entered the command SQUARE, the Turtle would respond with an error message, usually something like, "I DON'T KNOW HOW TO SQUARE." You must write a procedure for every new command you teach the Turtle.

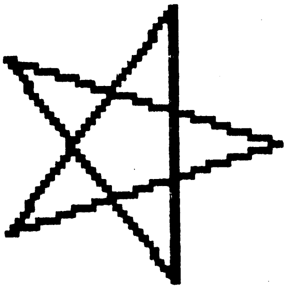
Writing a procedure is easy. Try this:

Take one of your favorite designs from Chapter One and think up a name for it. Any name you like is fine. Or you can follow the procedure for TO STAR below:

1. Type the word TO, followed by a space and the name of your procedure. The word TO tells the Turtle that you are writing a procedure. Press RETURN or ENTER.

> **APPLE and ATARI LOGO:** You will now see a symbol like this > to the left of the cursor. This means that you are now writing a procedure.
>
> **TI and T/K LOGO:** In these versions you will now enter the *Editor.* Your screen will change, and the name of the procedure you just typed in will appear in the upper left hand corner. Press RETURN or ENTER to bring the cursor to the next line.

2. On the next line under your procedure name, type the instructions for making your design. For example:



```
TO STAR
REPEAT 5 [FD 45 LT 144]
END
```

3. When you've finished typing in your instructions, type END on the last line. (NOTE: In **TI Logo,** END is already there, so you don't have to type it again. Make sure, though, that you don't have a blank line above the word END.)

4. To return to the graphics screen, do one of the following:

> **APPLE and TK LOGO:** Press CTRL and C to leave the Editor.
>
> **ATARI LOGO:** Press ESC.
>
> **TI LOGO:** Press FCTN and BACK (above the #9 key) to leave the Editor.

The Turtle now knows a new command—the name of your procedure. Congratulations! You are now a computer programmer!

To see the Turtle obey your new command, just type the name of your procedure followed by RETURN or ENTER. (Do not type the word TO here, just the name itself.) For example:

```
STAR (RETURN or ENTER)
```

When given the new command, the Turtle will automatically follow the instructions inside the STAR procedure, REPEAT 5 [FD 45 LT 144], and draw a star.

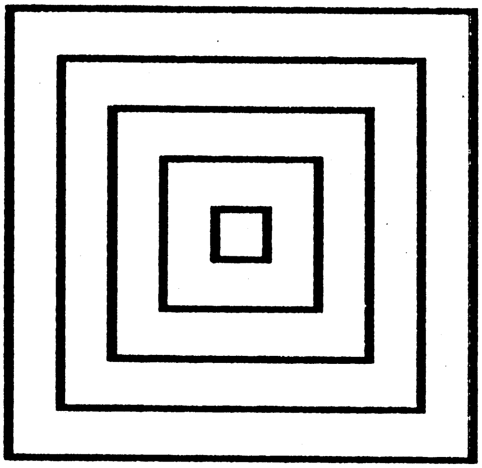
You can make up as many procedures as you want. Each time you name a procedure and teach the Turtle how to draw it, the Turtle will remember and draw the design every time you type in the procedure name. (Remember, though, that if you turn off the computer without saving your procedures first, the Turtle won't remember them the next time. For more about saving your work, see page 47.)

You can put as many instructions and Logo primitives into your procedures as you want. (A primitive is actually a small procedure.) For instance, if you have a design that works better with the Turtle hidden, the first instruction in your procedure might be HT. Or you could set the Turtle's position somewhere besides HOME to begin the drawing:

```
TO STAR
PU FD 50 PD
REPEAT 5 [FD 45 LT 144]
END
```

In this procedure every time you type STAR, the Turtle first moves 50 turtle steps without drawing a line (PU FD 50), then puts the pen down (PD) before following the rest of the instructions for drawing the STAR. (REPEAT 5 [FD 45 LT 144].)

If you want to try writing a longer procedure, see the following example of TO SQUARES and use it as a guide.



```
TO SQUARES
PD HT
REPEAT 4 [FD 90 RT 90]
```

```
FD 10 RT 90
PU FD 10 LT 90 PD
REPEAT 4 [FD 70 RT 90]
FD 10 RT 90
PU FD 10 LT 90 PD
REPEAT 4 [FD 50 RT 90]
FD 10 RT 90
PU FD 10 LT 90 PD
REPEAT 4 [FD 30 RT 90]
FD 10 RT 90
PU FD 10 LT 90 PD
REPEAT 4 [FD 10 RT 90]
PU HOME PD
END
```

MODES

Different versions of Logo have different ways of displaying things on the screen. These are called *modes.*

TI and **T/K** each have two modes: 1) The *command mode* or *toplevel* is where you type in commands and Logo carries them out. 2) The *edit mode* is where you write and change procedures.

Apple and **Atari** each have three modes: 1) The *command mode* is where you type in commands and Logo carries them out. 2) The *edit mode* is used for creating, changing, or correcting your procedures. 3) The *procedure definition mode,* which displays this symbol > after you type the word TO, is used for writing procedures.

EDITING, FIXING, AND CHANGING PROCEDURES

Fixing mistakes or changing things around in procedures is called *debugging* or editing. If you want to add something to a procedure or change something in it, you can use the Logo Editor.

To get into the Editor in **TI** and **T/K** Logo, type the word EDIT followed by a space and the name of your procedure. Then press RETURN or ENTER.

To get into the Editor in **Apple** and **Atari**, type EDIT or the abbreviation ED, followed by a space. Then type quotation marks ("), no space, and the name of your procedure. Then press RETURN or ENTER.

You are now in the edit mode.

To find out how to move the cursor around in your text, refer to the table of CTRL keys on the following pages.

Control (CTRL) Characters

APPLE LOGO

These CTRL characters can be used in both the command mode and the edit mode:

CTRL-A	Moves the cursor to the beginning of a line.
CTRL-B	Moves the cursor one space to the left.
CTRL-D	Deletes the character directly under the cursor.
CTRL-E	Moves the cursor to the end of a line.
CTRL-F	Moves the cursor one space to the right.
CTRL-G	Stops Logo from doing whatever it's doing in command mode. Returns to the command mode without recording any changes made in edit mode.
CTRL-K	Erases everything to the right of the cursor on a line.
CTRL-W	Stops text from scrolling off the top of the screen.
CTRL-Y	Prints a copy of the last line you typed.

These CTRL characters control what you see on your screen in the command mode:

CTRL-L	Works like FULLSCREEN to allow whole screen for graphics.
CTRL-S	Works like SPLITSCREEN to split screen for graphics and text.
CTRL-T	Works like TEXTSCREEN to allow full screen for text.

These CTRL characters work in the edit mode (the Editor) only:

CTRL-N	Moves the cursor down to the next line.
CTRL-P	Moves the cursor up to the previous line.
CTRL-O	Opens a line above the cursor when the cursor is at the beginning of a line.

ATARI LOGO

These CTRL characters can be used in both the command and the edit mode:

CTRL-A	Moves the cursor to the beginning of a line.
CTRL-DELETE BACK SPACE	Deletes character under the cursor.
CTRL-E	Moves the cursor to the end of a line.
CTRL-CLEAR	Erases everything to the right of the cursor.
CTRL←	Moves the cursor one space to the left.
CTRL→	Moves the cursor one space to the right.

CTRL-Y	Prints a copy of the last line you typed.
BREAK	Stops Logo from doing whatever it's doing in command mode. Returns to the command mode from the edit mode without recording any changes.

These CTRL characters control what you see on your screen in the command mode:

CTRL-F	Works like FS to allow full screen for graphics.
CTRL-S	Works like SS to allow split screen for graphics and text.
CTRL-T	Works like TS to allow full screen for text.

These CTRL characters work in the edit mode only:

CTRL-N or CTRL↓	Moves the cursor down to the next line.
CTRL-P or CTRL↑	Moves the cursor up to the previous line.
CTRL-INSERT	Opens a line above the cursor when the cursor is at the beginning of a line.

TI LOGO

These FCTN keys are printed above the number keys on your keyboard; the arrow keys are on the E, S, D, and X keys:

FCTN BEGIN	Moves the cursor to the beginning of a line.
FCTN PROC'D	Moves the cursor to the end of a line.
FCTN ↑	Moves the cursor up one line.
FCTN ↓	Moves the cursor down one line.
FCTN ←	Moves the cursor one space to the left.
FCTN →	Moves the cursor one space to the right.
FCTN CLEAR	Erases the character under the cursor and everything to the right of the cursor.
FCTN ERASE	Erases the character one space to the left of the cursor. Moves the whole line up if the cursor is at the beginning of a line.
FCTN DELETE	Erases the character above the cursor. Moves the line below up to that line if the cursor is at the end of a line.

In **TI**'s edit mode, the cursor moves one space to the right as you type each character.

TERRAPIN/KRELL LOGO

These CTRL characters can be used both in the command and the edit modes:

CTRL-A	Moves the cursor to the beginning of a line.
CTRL-B	Moves the cursor one space to the left.
CTRL-D	Deletes character directly under the cursor.
CTRL-E	Moves cursor to the end of a line.
CTRL-F	Moves the cursor one space to the right.
CTRL-K	Erases everything to the right of the cursor.

These CTRL characters control what you see on your screen in the command mode:

CTRL-F	Works like FULLSCREEN to allow whole screen for graphics.
CTRL-S	Works like SPLITSCREEN to split the screen for graphics and text.
CTRL-T	Works like TEXTSCREEN to allow whole screen for text.

These CTRL characters work in the edit mode (the Editor) only:

CTRL-N	Moves cursor down to the next line.
CTRL-P	Moves cursor up to the previous line.
CTRL-O	Opens a line above the cursor when the cursor is at the beginning of a line.

In all four versions of Logo, pressing RETURN or ENTER while in the edit mode does one of two things:

1. If the cursor is at the end of a line, a new line will open up below that line.
2. If the cursor is positioned somewhere within a line, the section of line to the right of the cursor will move down to the next line.

When you've finished changing or debugging your programs, you can get out of the edit mode by doing one of the following:

APPLE and T/K LOGO:

CTRL C	Returns to the command mode and records any changes made in your procedures while in the edit mode.
CTRL G	Returns to the command mode without recording any changes.

ATARI LOGO:

ESC	Returns to the command mode and records any changes made in your procedures while in the edit mode.

BREAK	Returns to the command mode without recording any changes.

TI LOGO:

FCTN BACK	Returns to the command mode and records any changes made while in the edit mode.

EDITING MORE THAN ONE PROCEDURE

If you want to edit more than one procedure, type EDIT or ED followed by a space.* Then list the procedures you want to edit within brackets. For example:

```
EDIT [STAR STARSHOW STARDESIGN] (RETURN)
```

The editor will now display all of the procedures listed within the brackets.

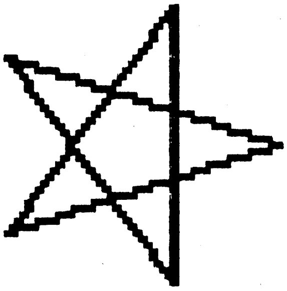
You can also write new procedures under the old ones in the Editor. Under the word END in the existing procedure, leave a blank line. Then type the word TO followed by a space and the new procedure name. Don't forget to type END when you have finished listing the instructions for your new procedure.

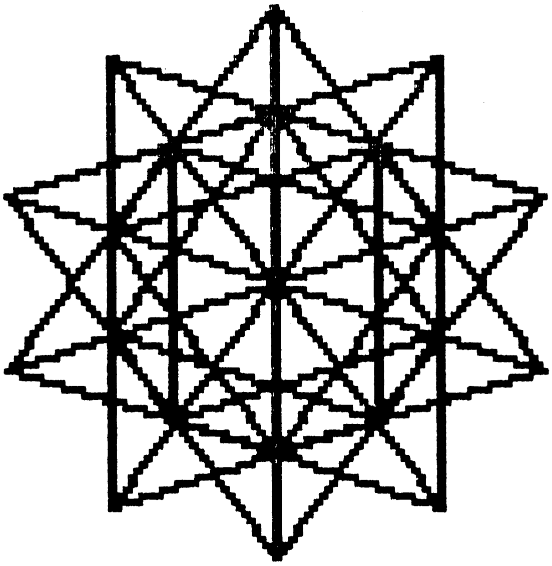
*You cannot edit more than one procedure at a time in **TI Logo.**

NESTING PROCEDURES

Once you have named a new procedure and given the Turtle instructions for carrying it out, your procedure becomes a new command for the Turtle. You have taught it a new word. Your new procedures or commands can be used as parts of other procedures to make more complicated designs and drawings. This is called *nesting procedures*. Take our STAR procedure (see page 32). You don't need to draw a whole new star every time you want to include a star in one of your programs—simply include it as part of your procedure.

For example, we have put STAR inside the two procedures below to create new designs:





```
TO STARDESIGN
REPEAT 10 [STAR RT 36]
END
```

STARDESIGN makes a circle of 10 STARS. The Turtle draws the original STAR, then turns right 36 degrees and draws another STAR. This is REPEATed 10 times to make STARDESIGN:

```
TO STARSHOW
STARDESIGN
LIGHTSHOW
END
```

Now you know that the Turtle already knows how to STARDESIGN, but we must define LIGHTSHOW before the Turtle can complete the above program. LIGHTSHOW changes the background color of your monitor several times. (For more on color, see Chapter Three.)

APPLE and ATARI LOGO:

```
TO LIGHTSHOW
REPEAT 10 [SETBG 2 SETBG 3 SETBG 4 SETBG 5]
END
```

TI LOGO:

```
TO LIGHTSHOW
REPEAT 25 [CB 2 CB 3 CB 4 CB 5]
```

T/K LOGO:

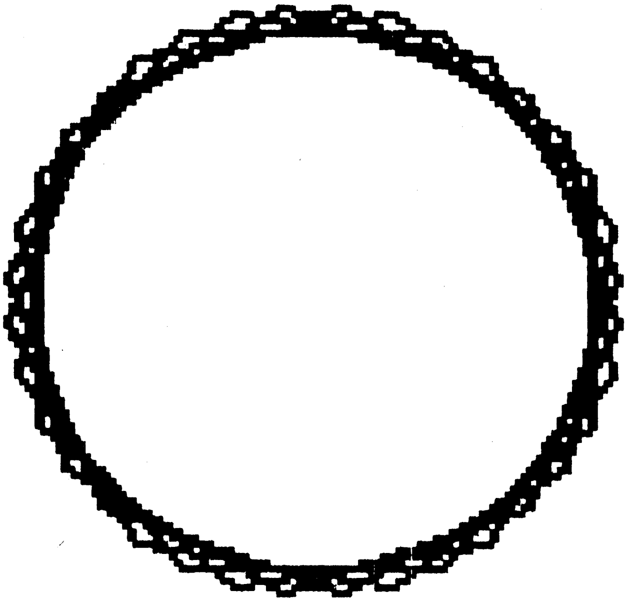
```
TO LIGHTSHOW
REPEAT 10 [BG 2 BG 3 BG 4 BG 5]
END
```

Procedures can be used by themselves or as parts of other, larger procedures. If you take another look at STARSHOW you can see that it has three procedures nested inside of it: STAR (inside of STARDESIGN), STARDESIGN, and LIGHTSHOW. When a procedure is used as part of a larger procedure, it is called a *subprocedure.*

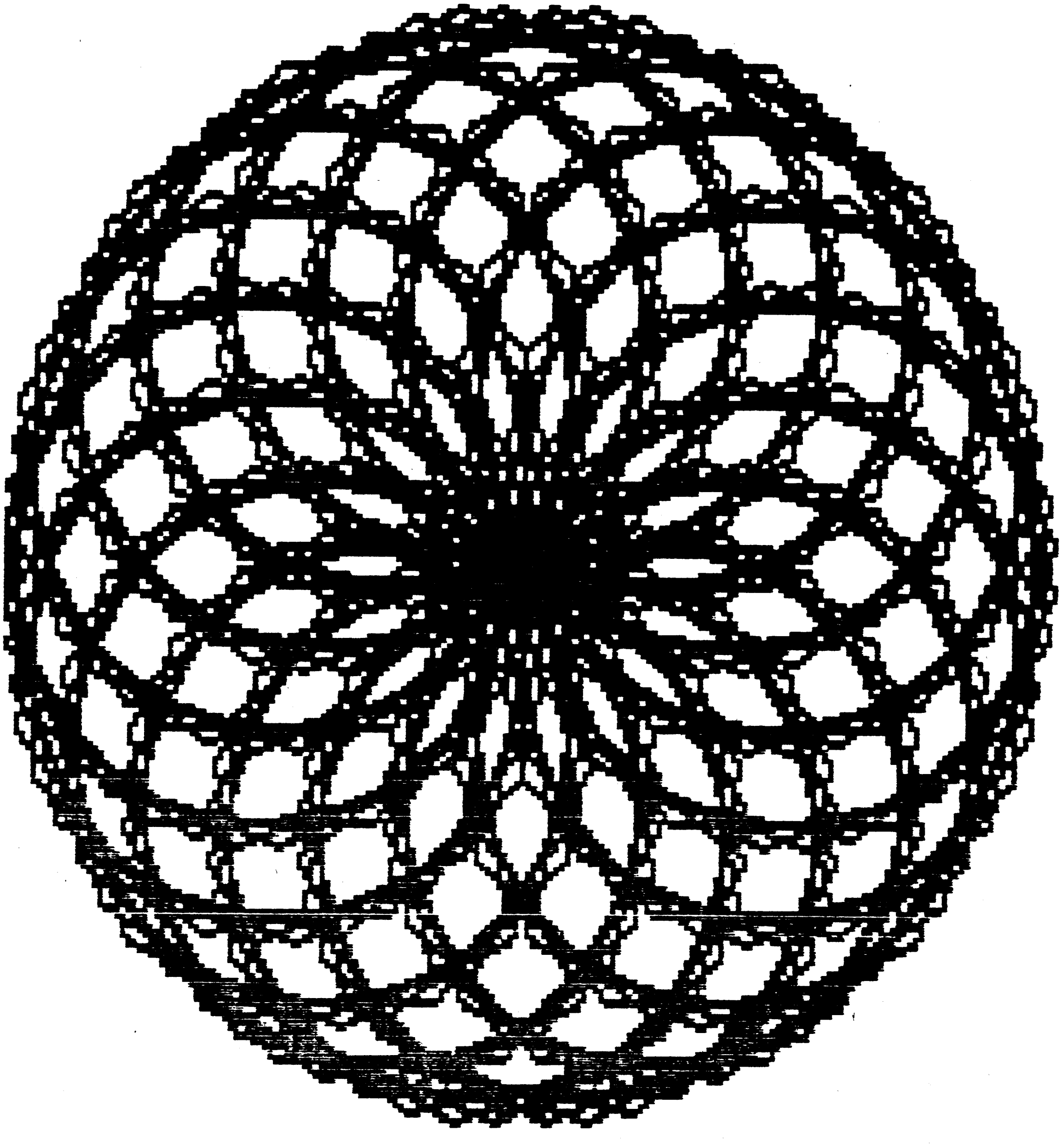
On the following pages are some programs written by kids that include subprocedures.

Flower Circle

This procedure uses REPEAT and a simple subprocedure called CIRCLE. Try experimenting with this program and some of the color commands in the following chapter.



```
TO CIRCLE
REPEAT 50 [FD 50 RT 50]
END
```



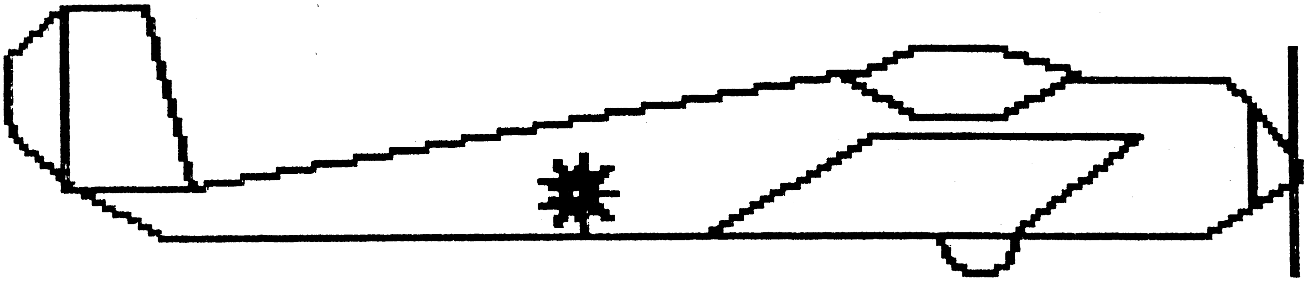
```
TO FLOWER.CIRCLE
PU BK 20 PD HT
REPEAT 18 [CIRCLE FD 20]
END
```

Clint's Plane

This program was written by Clint T. of San Francisco, California. In this procedure RD stands for READY. RD1, RD2, and RD3 are subprocedures for positioning the Turtle to draw separate parts of the PLANE. (This program was written for **Apple Logo**.) See also color insert, page 60.

```
TO PLANE
RD1
TAIL
HULL
STAR 15
RD2
PROP
RD3
WHEEL
END

TO RD1
HT LT 90
PU FD 100 RT 90 PD
END

TO TAIL
LT 50 FD 15 RT 40 FD 3 RT 8 FD 12 RT 50
FD 15 RT 132 FD 36 BK 36 LT 90 FD 16 RT 75
FD 37 RT 105 FD 23
END

TO HULL
LT 150 FD 20 LT 30 FD 200 LT 30 FD 20 BK 8 LT 60 FD 20
RT 140 FD 14 RT 40 FD 4 BK 4 LT 40 BK 14 RT 170 FD 7
LT 40 FD 30 LT 25 FD 15 RT 25 FD 15 RT 25 FD 15
RT 130 FD 15 RT 25 FD 15 RT 15 FD 15
PU
BK 15 LT 15 BK 15 LT 15 BK 15
PD RT 185 FD 130
PU
LT 80 FD 50 LT 90 FD 100 LT 90 FD 40 RT 60 PD
FD 40 RT 30 FD 50 RT 140 FD 30
END

TO STAR :SIZE
PU HOME FD 5 PD
REPEAT 10 [LT 160 FD :SIZE]
END

TO RD2
PU FD 5
LT 110 FD 144 LT 90
PD
END
```

```
TO PROP
FD 35 BK 40 FD 45
END

TO RD3
PU LT 90 FD 54
LT 90 FD 38 PD
END

TO WHEEL
ARCR 7 180
END
```

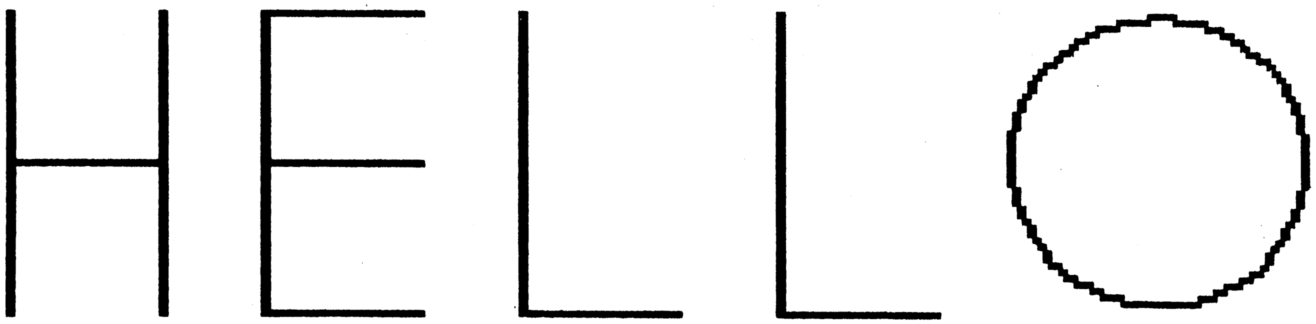


Hello

(See also color insert, page 61.)

```
TO HELLO
HT SETUP
H
MOVE
E
MOVE
REPEAT 2 [L MOVE]
PU SHORT PD
O
END

TO SHORT
FD 30
END

TO H
LONG
LT 180
REPEAT 2 [SHORT LT 90]
SHORT LT 180 LONG LT 90
END

TO E
LONG RT 90
SHORT LT 180
REPEAT 2 [SHORT LT 90]
SHORT LT 180
REPEAT 2 [SHORT LT 90]
SHORT
END

TO SETUP
PU LT 90 FD 120 RT 90 PD
END

TO LONG
FD 60
END

TO MOVE
PU FD 15 PD LT 90
END
```

```
TO L
LONG RT 180
LONG
LT 90
SHORT
END

TO O
REPEAT 90 [FD 2 RT 4]
END
```



WHAT TO DO AT THE END OF A LOGO SESSION

When you turn off your computer, all your designs and procedures will be lost forever unless you SAVE them first. Several methods for saving your work will be outlined below. You should make sure, however, that you're only saving what you need to save.

Looking Things Over

As you write and edit procedures, your computer stores them in its memory. The memory is also known as your workspace. Here are the commands for looking over what is in your workspace.

APPLE and ATARI LOGO:

POTS (Print Out Titles)—Prints out names of all the procedures you've entered in your workspace during your Logo session.

PO (Print Out)—This command must be followed by a space, quotes ("), and the name of a procedure. Prints out the contents of a procedure. PO followed by brackets and a list of procedure names will print out the contents of all the procedures listed within the brackets.

POPS (Print Out Procedures)—Prints out the contents of all procedures in your workspace.

POALL (Print Out All)—Prints out the contents of all procedures in your workspace and the value of all variables.

BUGBUSTER—To keep the text from scrolling off your screen before you've had a chance to look at it, use the command CTRL W or CTRL 1.

TI LOGO:

PP (Print Procedures)—Prints out the names of all the procedures you've entered during your Logo session.

PO (Print Out)—This command must be followed by the name of a procedure. Prints out each step of a procedure.

PA (Print All)—Prints out the names of all of your procedures and their contents.

TERRAPIN / KRELL LOGO:

PO (Print Out)—This must be followed by a space and the procedure name. PO will also print out lists of procedures enclosed in brackets.

POTS (Print Out Titles)—Prints out the names of all the procedures you've entered in your workspace during your Logo session.

PO ALL (Prints Out All)—Prints out titles and contents of all procedures entered in your workspace during your Logo session.

Taking Out The Garbage

There's no point in saving things you don't like or want, so after you've looked over what you have, it's time to take out the garbage. You can change any procedure by using the Editor, and you can erase any procedures you don't want by using the following commands:

APPLE and ATARI LOGO:

ERASE or ER—Followed by a space and quotes (") and the name of the procedure you want to erase. This command can also be used with brackets and a list of procedures to erase more than one procedure at a time.

ERALL (Erase All) or ERPS (Erase Procedures)—This erases all the procedures stored in your workspace.

TI LOGO:

ERASE—Followed by a space and the name of the procedure you want to erase.

ERASE ALL—Erases all procedures stored in your workspace.

TERRAPIN/KRELL LOGO:

ERASE—Followed by a space and the name of the procedure you want to erase.

ERASE ALL (or ER ALL)—Erases all procedures stored in your workspace.

HOW TO SAVE YOUR WORK

After cleaning up your workspace, it's time to SAVE what's left. There are four ways to save your work: Write it down, save on cassette tapes, save on disks, and print out your procedures. Each of these methods for saving your work are discussed on the following pages.

1. Write your procedures down in a journal or workbook. This is a good method of keeping track of your Logo programs, because a written record provides a quick and easy reference for debugging, or in case something happens to your disks or cassette tapes.

2. **TI** and **Atari Logo** procedures can be saved on cassette tape. Sprites and Tiles can be saved in **TI Logo**, and Shapes can be saved on **Atari**. (For more on Sprites, Tiles, and Shapes, see Chapter 5.) The directions for saving on cassette tape are somewhat different for each system. Consult your manual for specific instructions on how to save on cassette tapes.

3. Most systems also enable you to save your work on disks, providing you have a disk drive attached to your computer. To save your work on a disk, you will first have to format or initialize your diskette (set up the disk to run on your computer system); consult your DOS manual, or your owner's manual for instructions. For each version, consult your Logo manual for complete instructions on how to transfer your work to a disk.

APPLE: You can SAVE procedures.

ATARI: You can SAVE both procedures and shapes.

TI LOGO: You can SAVE procedures, Tiles and Shapes on a diskette.

TERRAPIN / KRELL: You can SAVE procedures and pictures that are not in procedures.

4. Most systems will allow you to print out procedures and graphics if you have a printer attached to your computer. Check your owner's manual for both your computer and your printer for instructions on printing out graphics and procedures.

General Rules for Saving Your Work

Although you should be sure to look in your Logo manual for specific methods for saving your work, there are some general rules that apply to all systems:

1. When saving work on a disk or cassette tape, you are creating a file of the procedures contained in your workspace. When you name a file and save it, you are saving all the procedures in your workspace. That's why it's a good idea to clean up your workspace before attempting any saving operation. It avoids confusion later on when you recall your files, and saves space on your disks and cassette tapes.

Your Logo manual will list specific commands for saving your work and for recalling your files later on. Listed below are some of the most frequently used commands for each system so you will have an easier time looking them up.

APPLE LOGO:

SAVE—Typing SAVE, a space, a ", and a file name and pressing RETURN will save your file to disk.

LOAD—LOAD, a space, a ", and a file name will read that file back into your computer's memory from a disk.

CATALOG—The CATALOG command will list all of the file names on a disk.

ERASEFILE—ERASEFILE followed by a space, a ", and a file name will erase that file from your disk. To erase more than one file, type ERASEFILE followed by a space and a list of file names in brackets.

ATARI LOGO:

SAVE "C: —Followed by a file name will save procedures on a cassette tape.

SAVE "D: —Followed by a file name will save procedures on a disk.

SAVE "P: —Followed by a file name will print out contents of workspace.

CATALOG "D: —Lists the file names stored on a disk.

LOAD "D: —Followed by a file name loads that file from a disk into your computer.

LOAD "C: —Followed by a file name loads that file from a cassette tape into your computer.

ERF "D: —Followed by a file name erases that file from a disk.

TI LOGO:

SAVE—Typing SAVE and pressing the ENTER key will put you in **TI**'s SAVE mode.

RECALL—Typing RECALL and pressing ENTER will put you in **TI Logo**'s RECALL mode so you can recall your files and read them back into your computer.

TERRAPIN / KRELL LOGO:

SAVE—Followed by a space and a file name saves that file to a disk.

READ—Followed by a space and a file name reads that file into your computer.

CATALOG—Lists the file names stored on a disk.

ERASEFILE—Followed by a space and a file name erases that file from a disk. To erase more than one file, type ERASEFILE followed by a space and a list of file names in brackets.

BUGBUSTER: If you use the same file name twice in **Texas Instruments, Terrapin/Krell,** or **Atari Logo** the old file bearing that name will automatically be erased and a new one created. So be careful when naming your files.

WORDS TO REMEMBER: CHAPTER TWO

PROCEDURE—A procedure is a computer program that teaches the Turtle new commands. Procedures consist of three parts: NAME, SERIES OF INSTRUCTIONS, and END.

DEBUGGING—Fixing or changing your procedures, getting the "bugs" out.

COMMAND MODE—The mode where you give commands and the Turtle draws them on your screen. This is sometimes known as TOPLEVEL.

EDIT MODE—The mode used to edit procedures and in some versions of Logo to write them as well.

PROCEDURE DEFINITION MODE—In some versions of Logo this is a mode used to write procedures.

NESTING PROCEDURES—Combining procedures to build more complex procedures.

SUBPROCEDURE—A procedure contained within another procedure.

3

All About Color

Each version of Logo has color commands. If your computer is hooked up to a color TV or color monitor, you will be able to use Logo's color commands to create special effects in a variety of colors.

If you're working with a monitor or TV that doesn't have color, you will still be able to use the color commands. Flashing lines and turtle lines with different thicknesses can be produced on a green screen monitor or a black and white TV. Read through the following section on color for your version of Logo and try some examples. See the kinds of things that are possible using Logo color and go on from there to create your own special effects using color commands.

USING COLOR COMMANDS

To use color commands you will need a primitive and a color code number as an input. There are commands for setting both the background color of your screen monitor and the pencolor—the color of the lines you want the Turtle to draw. Colors are entered by number in Logo, so check the charts below for the primitives and color code numbers for your version of Logo.

APPLE LOGO

Primitives

SETBG—Is the command for setting the screen background color.
SETPC—Is the command for setting the pencolor.

Color Code Numbers

Apple Logo has six colors available. Each color has a code number assigned to it:

BLACK	0	
WHITE	1	
GREEN	2	
VIOLET	3	
ORANGE	4	
BLUE	5	
BLACK	6	(SETBG only)

For example, to set the background color of the screen monitor to blue, you type SETBG 5, followed by a RETURN. To change it to orange, type SETBG 4 (RETURN). To set the color of the Turtle's lines to green, type SETPC 2. How would you set the pencolor to white?

The **Apple** computer color system affects the way some background and pencolors work together. Because of this, certain pencolors do not show up against certain background colors, while other pencolors show up as different colors than their code number colors. Below is a list of the six background colors and how they interact with some pencolors. Remember that no matter what version of Logo you are using, if you enter a pencolor that is the same as your background color, it won't show up.

BG 0—All pencolors show up as coded.
BG 1—All pencolors show up as coded.

WON'T SHOW UP		SHOWS UP AS DIFFERENT COLOR
BG 2	PC 2,4	PC 5 is violet
BG 3	PC 3,5	PC 2 and 4 are green
BG 4	PC 2,4	PC 3 is blue
BG 5	PC 3,5	PC 2 is orange
BG 6	PC 0	PC 4 and 1 are blue
		PC 2 is violet

BUGBUSTER—Background color 6 tends to give inconsistent pencolors. It is best used with non-color monitors. SETBG 6 makes all turtle lines appear thinner than other background colors.

ATARI LOGO

Primitives

SETBG—Is the command for setting the background color.

SETC—Is the command for setting the color of a Turtle or Shape (see page 94).

There are three pens available for drawing in **Atari Logo.** They are numbered 0, 1, and 2. Before setting a pencolor in **Atari Logo,** you must first set the pen number you want to use. The primitives for setting pen numbers and pencolors are below:

SETPN—Is the command for setting the pen number. It is followed by an input number of 0, 1, or 2.

SETPC—Is the command for setting the pencolor and takes 2 numbers as input; the first is the number of the pen, and the second is the number of the color you want to use.

If you want to draw with more than one pen at once, you may put more than one pen number in brackets. For example, SETPN [0 1 2] sets all three pens. If you want different portions of a design drawn in different colors, you must change pens first before changing the pencolor. Otherwise, the pencolor for your entire design will change automatically.

Color Code Numbers

There are 128 colors available in **Atari Logo.** The chart below lists the colors and their code numbers:

0- 7	GRAY
8- 15	LIGHT ORANGE (GOLD)
16- 23	ORANGE
24- 31	RED-ORANGE
32- 39	PINK
40- 47	PURPLE
48- 55	PURPLE-BLUE
56- 63	BLUE
64- 71	BLUE
72- 79	LIGHT BLUE
80- 87	TURQUOISE
88- 95	GREEN-BLUE
96-103	GREEN
104-111	YELLOW-GREEN
120-127	LIGHT ORANGE

These are actually made up of 16 **Atari** colors. Each one has 8 shades. For each color, the lowest code number is the darkest shade of that color and the highest number is the lightest shade of that color. For example, 0 is black and 7 is white.

TI LOGO

Primitives

COLORBACKGROUND or CB—Is the command for setting background color.

SETCOLOR or SC—Is the command for setting the color of the Turtle's lines and Sprites and Tiles.

Color Code Numbers

There are 16 colors available in **TI Logo.** Each color has a color code number assigned to it. Check the chart below for the colors and their code numbers:

CLEAR	0	RUST	8
BLACK	1	ORANGE	9
GREEN	2	YELLOW	10
LIME	3	LEMON	11
BLUE	4	OLIVE	12
SKY	5	PURPLE	13
RED	6	GRAY	14
CYAN	7	WHITE	15

You can also use the names of the colors themselves to set color with **TI Logo.** If you decide to set colors this way, the color name will need a colon (:) directly in front of it. For example:

CB :WHITE or SC :GREEN

TERRAPIN/KRELL LOGO

Primitives

BACKGROUND or BG—Is the command for setting the screen background color.

PENCOLOR or PC—Is the command for setting the pencolor.

Color Code Numbers

Terrapin/Krell Logo uses the same six colors as **Apple Logo.** The code numbers are also the same as for **Apple Logo.** Be sure to include a code number following your color primitives.

Using PC 6 in **Terrapin/Krell Logo** works like the PENREVERSE (See p. 55.) command in other versions of Logo. PC 6, followed by an input number, erases any lines it passes over or retraces, and draws new lines where there weren't any before.

USING COLOR

The possibilities for wonderful color effects in Logo programs are as endless as your own imagination. If you would like a particular design drawn in the same colors every time, you can enter color commands right into your procedures. To get an idea of how different colors work with your version of Logo, try drawing a design on a black background and then changing the background for different effects. In most versions of Logo, pencolors can also be changed.

REPEAT

Color commands can also be used in combination with other Logo primitives, as in TO LIGHTSHOW.

```
TO LIGHTSHOW
REPEAT 10 [SETBG 2 SETBG 3 SETBG 4 SETBG 5]
END
```

Remember to use the correct commands for your version of Logo. The same program for **Terrapin/Krell Logo** would look like this:

```
TO LIGHTSHOW
REPEAT 10 [BG 2 BG 3 BG 4 BG 5]
END
```

This will change the background color 40 times very fast. (10 REPEATs × 4 colors.)

Your choice of colors and the number of times you repeat the procedure above will differ depending on which version of Logo you are using. Having less than 10 REPEATS with **TI Logo** causes the colors to flash by so fast that much of the flashing effect is lost. Using too many of the same colors from one color group in **Atari Logo** will simply cause the screen to change shades rather than give a flashing effect. Keep experimenting with colors to find out what works best for your procedures.

PENREVERSE

APPLE LOGO: PENREVERSE or PX

ATARI LOGO: PX

TI LOGO: PENREVERSE or PR

T/K LOGO: Uses PENCOLOR 6 instead of PENREVERSE

PENREVERSE commands cause the Turtle to erase any lines on the screen that it passes over and draw lines where there weren't any before. After using a PENREVERSE command, you can restore your pen to normal drawing by typing the PENREVERSE command again in **Apple** and **Atari Logo**. For **TI Logo**, type PENDOWN or PD. (For **T/K Logo,** change the PENCOLOR.)

WAIT

WAIT is a command that can be very useful when used in combination with color commands. WAIT tells Logo to stop whatever it's doing and wait for as long as you tell it to before continuing on to the next command. Logo WAITs in 60ths of a second.

```
WAIT 1 = 1/60 second
WAIT 60 = 1 second
WAIT 300 = WAIT 5 seconds
```

Take the flashing screen procedure we were using before and insert a WAIT command.

```
TO FLASH 2
REPEAT 10 [SET BG 2 WAIT 60 SETBG 3 WAIT 60 SETBG 4 WAIT 60
SETBG 5 WAIT 60]
END
```

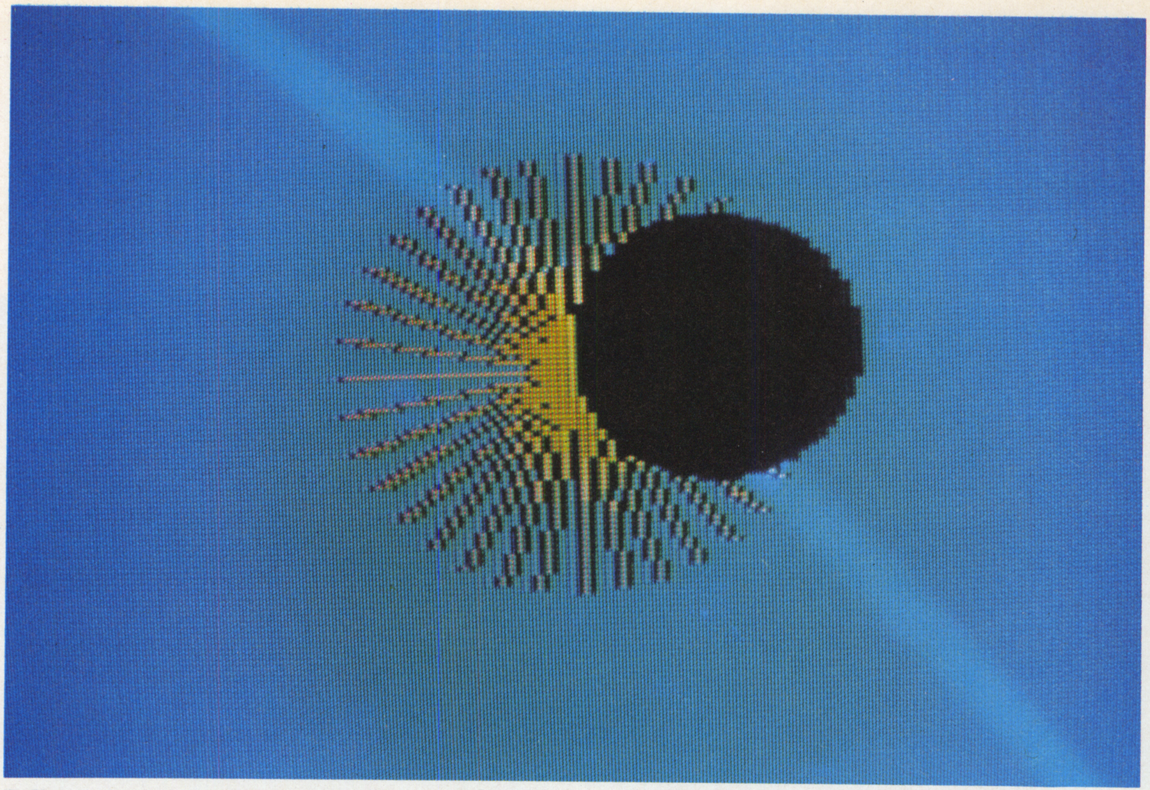
This causes the flashing screen to pause one second between colors. Try this:

Using the STAR procedure or one of your own designs, add a line using both color commands and the WAIT command. Vary the amount of time in WAIT and see what happens.

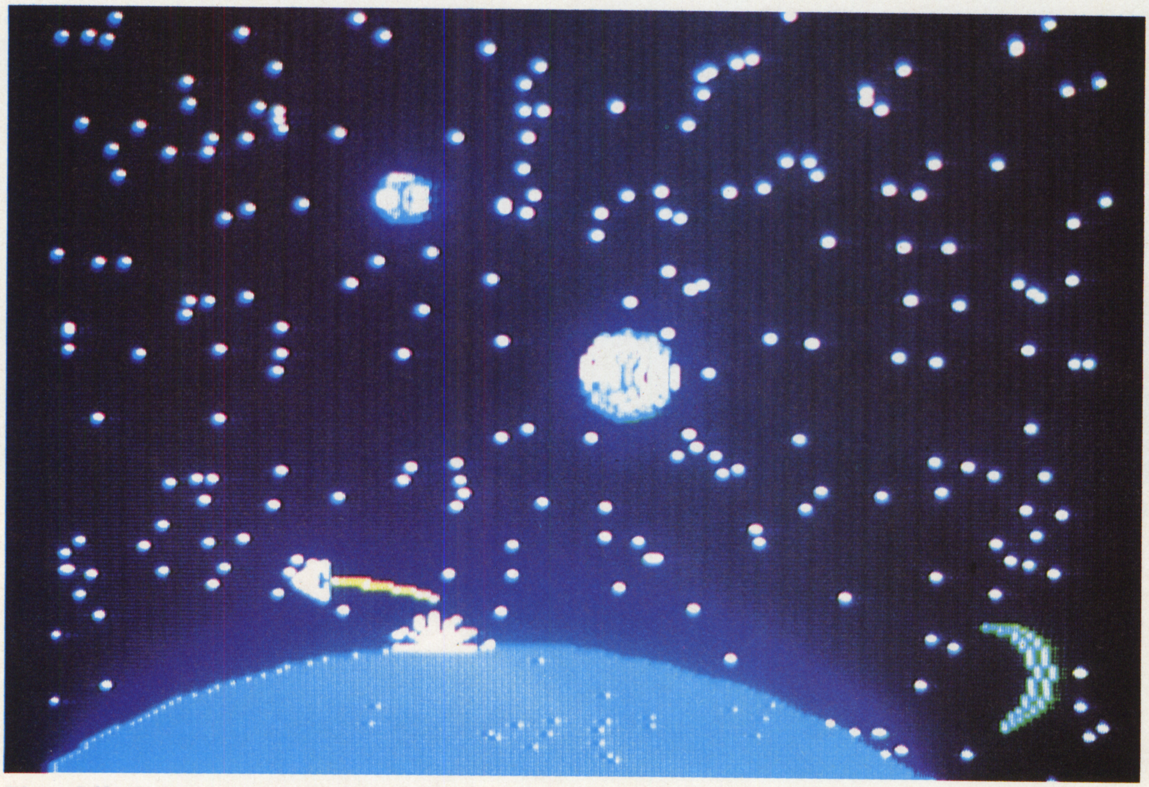
COMMANDS TO REMEMBER: CHAPTER THREE

PRIMITIVE	ABBREVIATION	INPUTS NEEDED	FUNCTION	VERSION
SETBACKGROUND	SETBG	COLOR CODE NUMBER 0–6	SETS BACKGROUND COLOR	APPLE
SETPENCOLOR	SETPC	COLOR CODE NUMBER 0–5	SETS COLOR OF LINES DRAWN BY THE TURTLE	APPLE
BACKGROUND	BG	COLOR CODE NUMBER 0–6	SETS BACKGROUND COLOR	TERRAPIN/ KRELL
PENCOLOR	PC	COLOR CODE NUMBER 0–6	SETS COLOR OF LINES DRAWN BY THE TURTLE	TERRAPIN/ KRELL

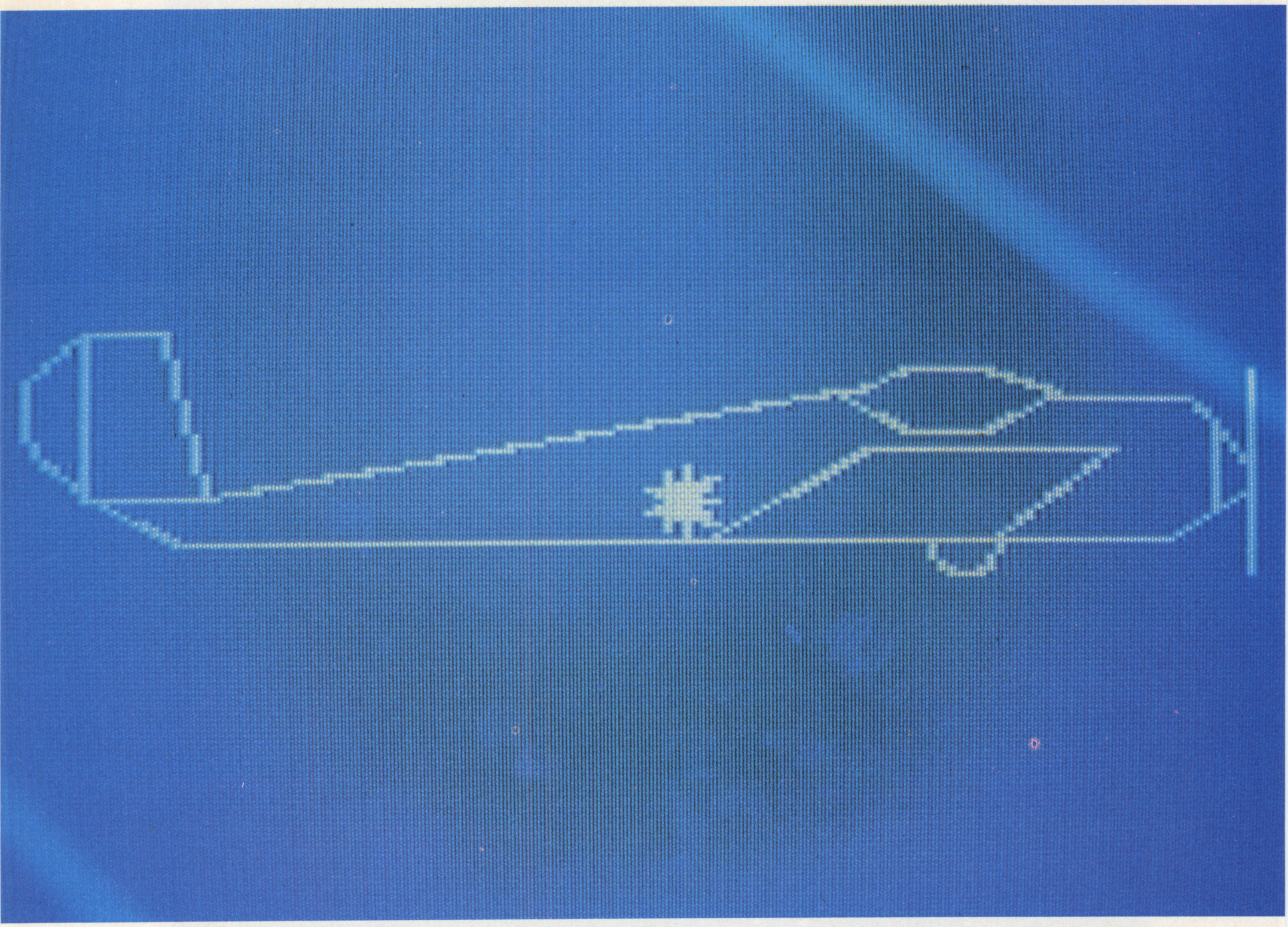
PRIMITIVE	ABBREVIATION	INPUTS NEEDED	FUNCTION	VERSION
COLORBACKGROUND	CB	COLOR CODE NUMBER 0–15 or COLOR NAME	SETS BACKGROUND COLOR	TEXAS INSTRUMENTS
SETCOLOR	SC	COLOR CODE NUMBER 0–15 or COLOR NAME	SETS COLOR OF SPRITES AND TILES AND TURTLE LINES	TEXAS INSTRUMENTS
SETBG	SETBG	COLOR CODE NUMBER 0–127	SETS BACKGROUND COLOR	ATARI
SETC	SETC	COLOR CODE NUMBER 0–127	SETS COLOR OF A TURTLE OR SHAPE	ATARI
SETPN	SETPN	PEN NUMBER 0, 1, or 2	SETS PEN NUMBER	ATARI
SETPC	SETPC	PEN NUMBER AND COLOR CODE NUMBER	SETS PEN COLOR	ATARI
PENREVERSE	PX (PR in TI LOGO)	NONE	REVERSES COLOR OF LINES IT PASSES OVER	APPLE
WAIT	NONE	NUMBER IN 60ths OF A SECOND	STOPS A PROGRAM FOR DURATION OF INPUT	ALL



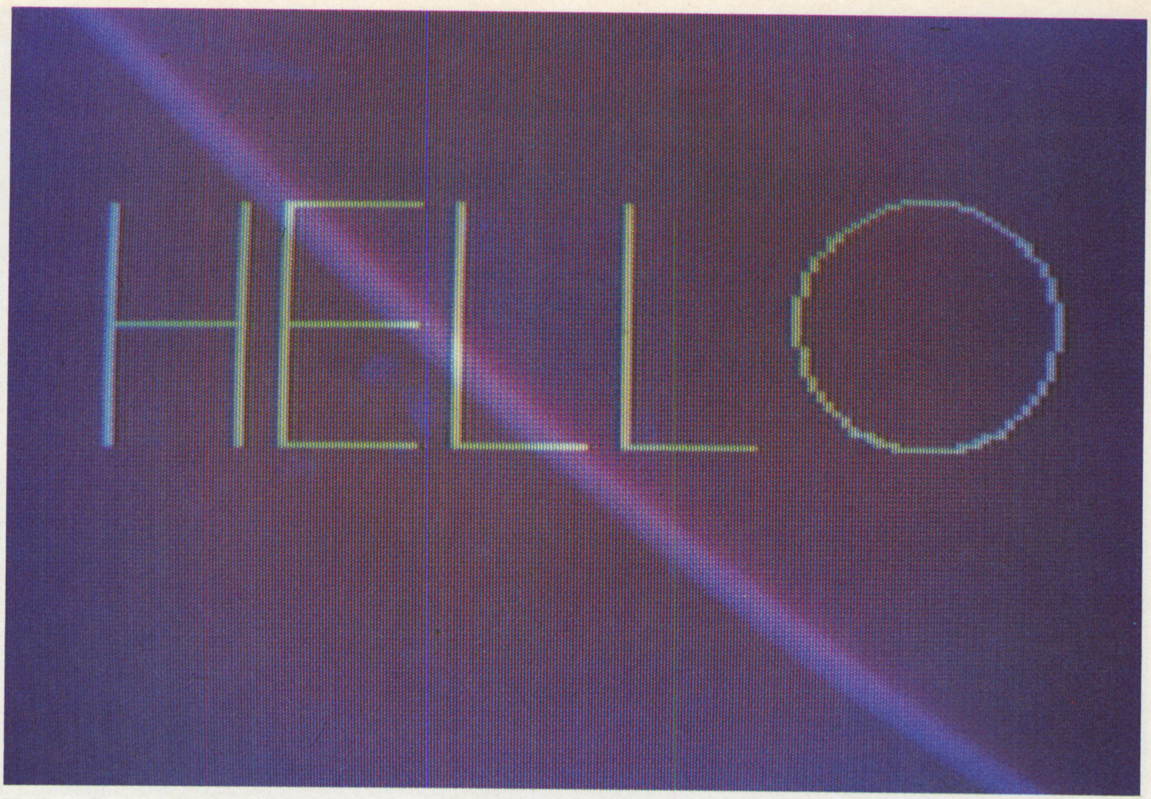
Eclipse *See procedure on page 24.*



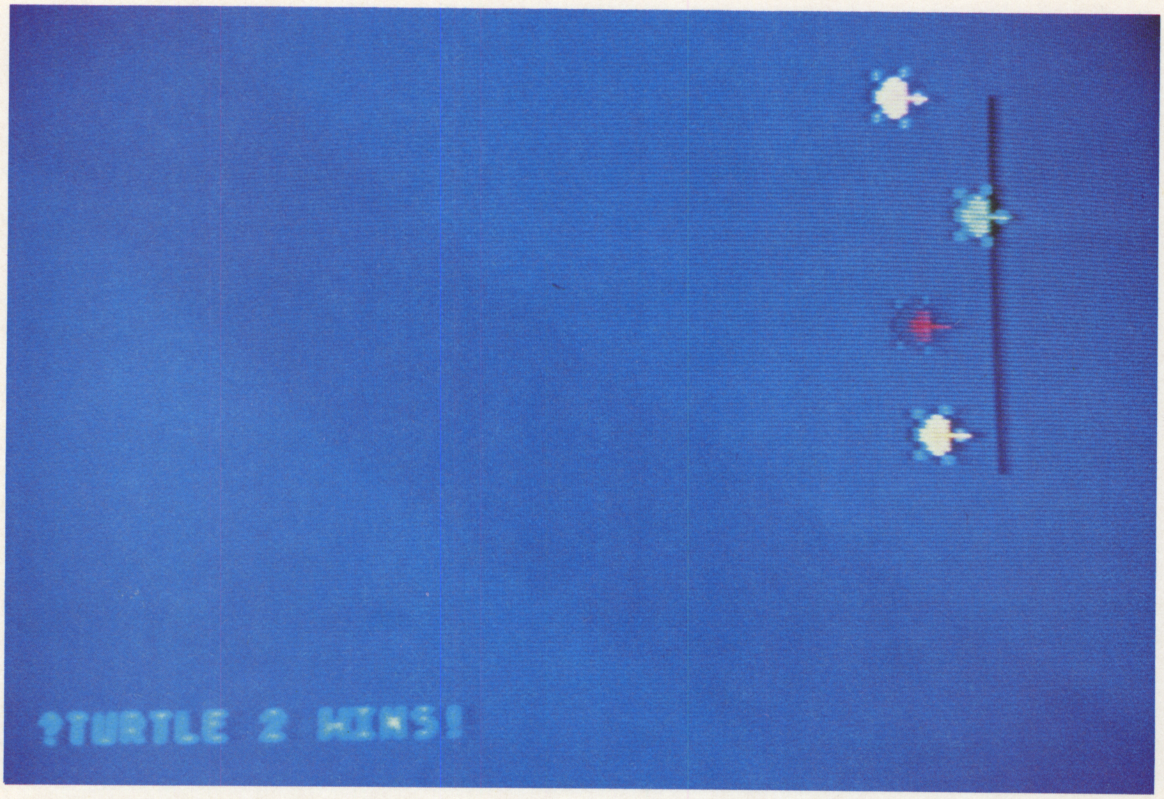
Blast Off *See procedure on page 107.*



Clint's Plane *See procedure on page 42.*

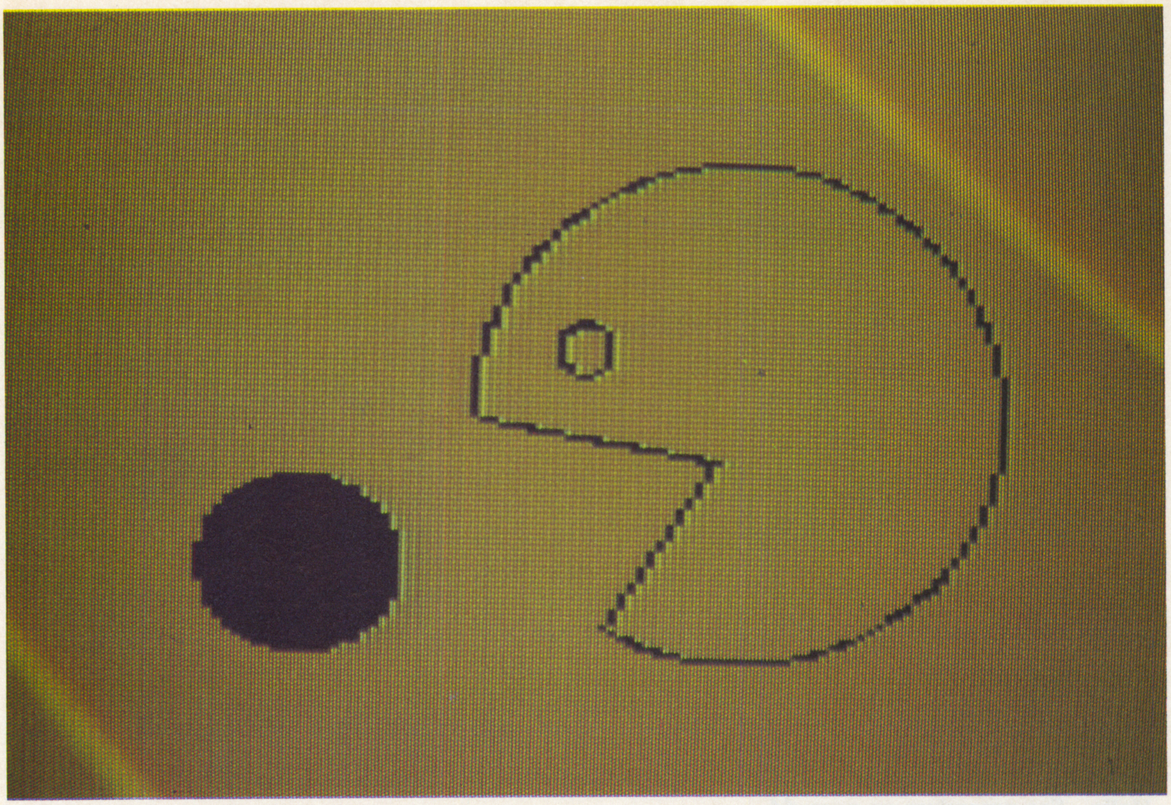


Hello *See procedure on page 44.*

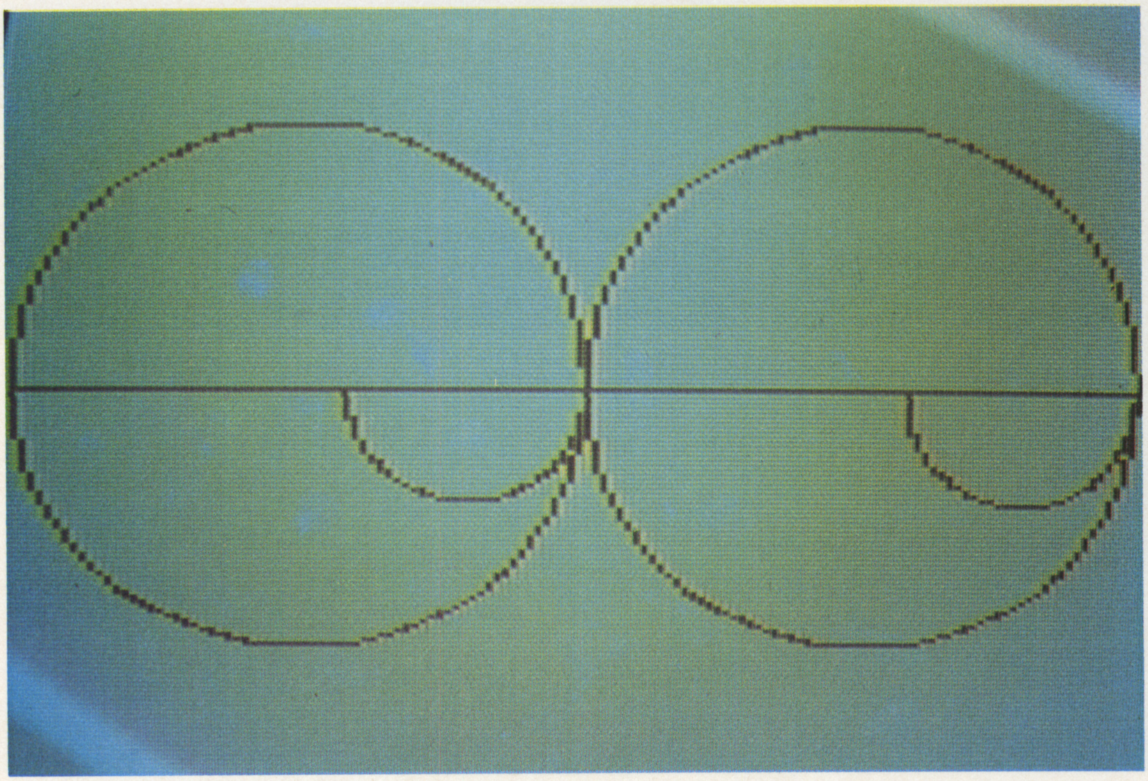


```
?TURTLE 2 WINS!
```

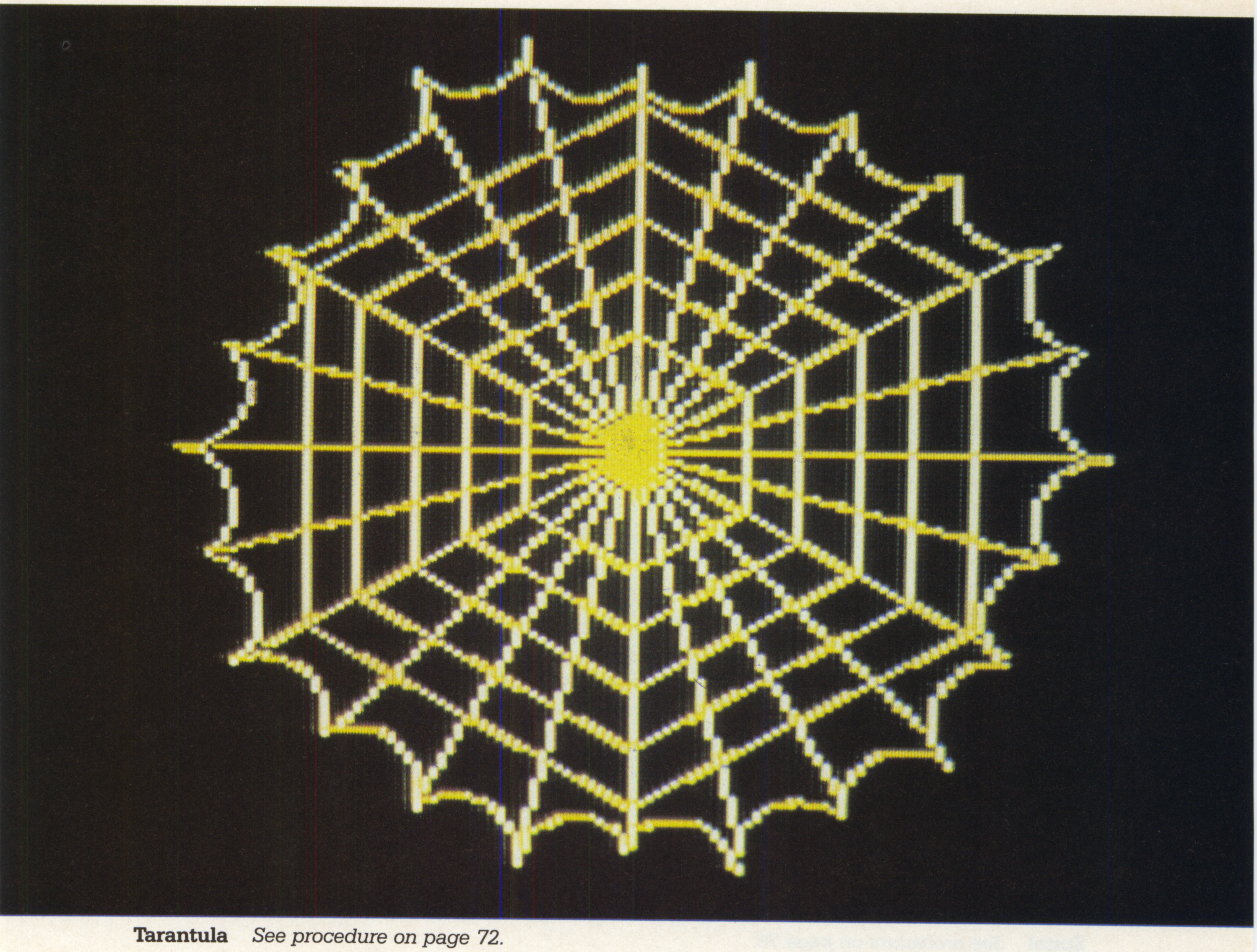
Andy's Turtle Race *See procedure on page 102.*



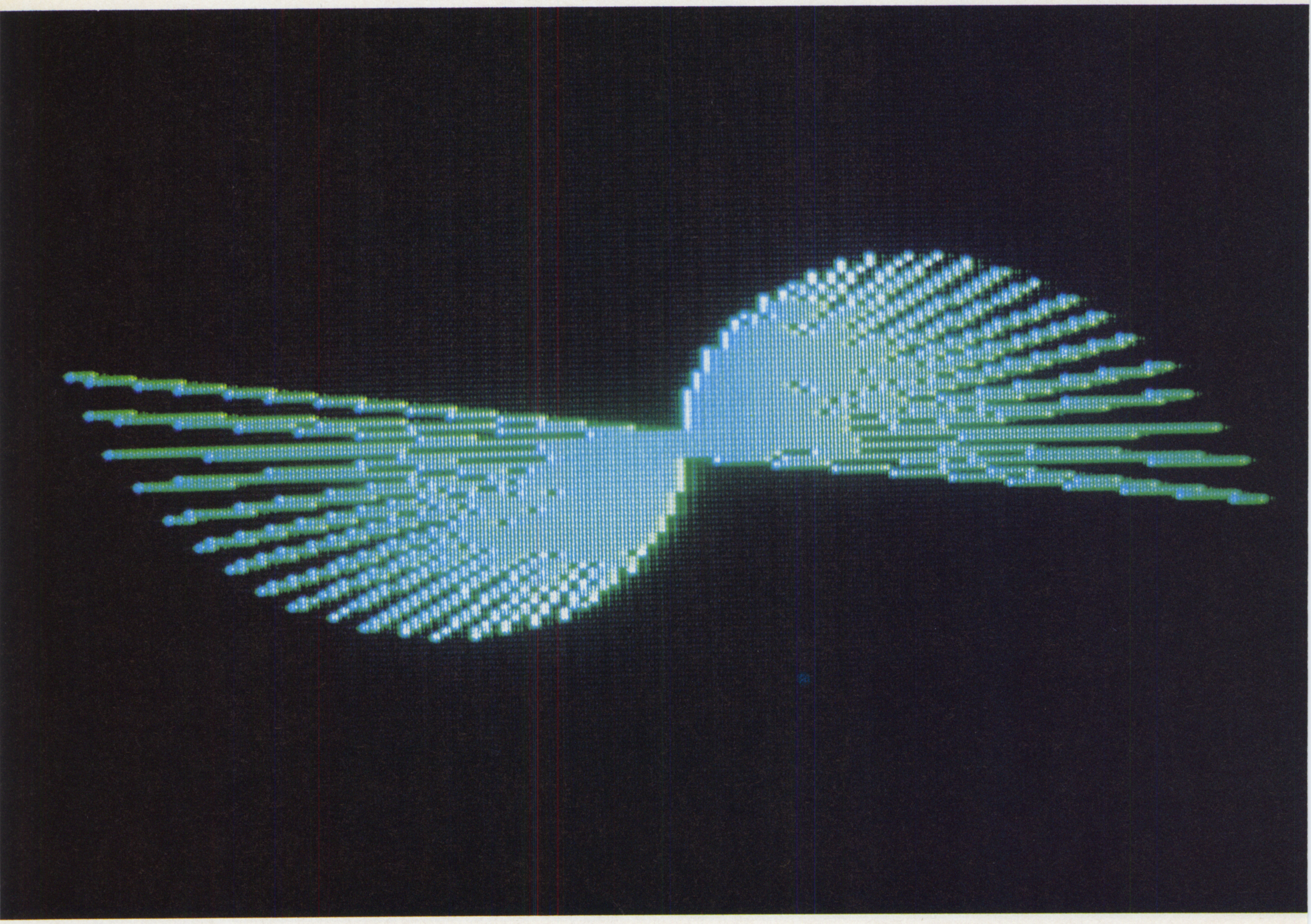
Pac Man *See procedure on page 69.*



Garfield's Eyes *See procedure on page 70.*



Tarantula *See procedure on page 72.*



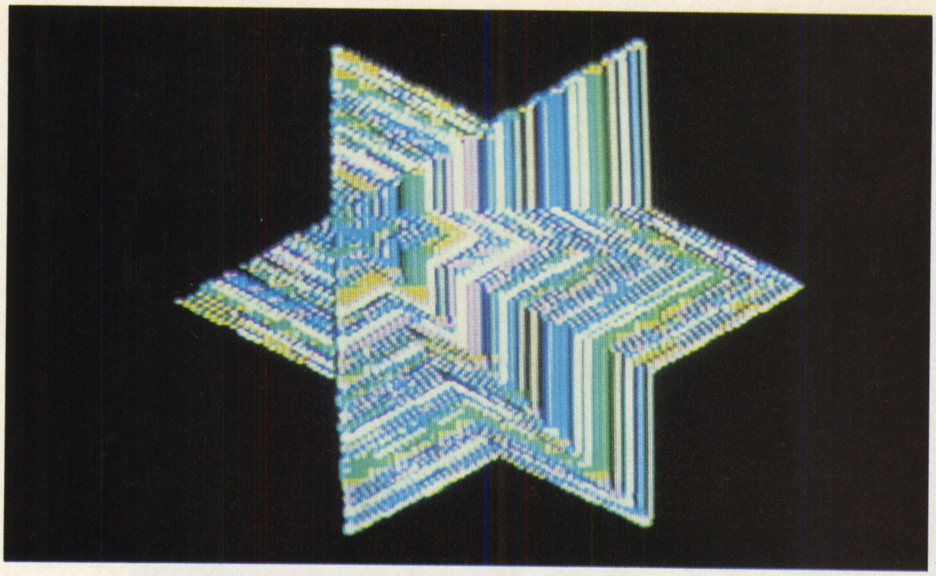
Spiral *See procedure on page 78.*

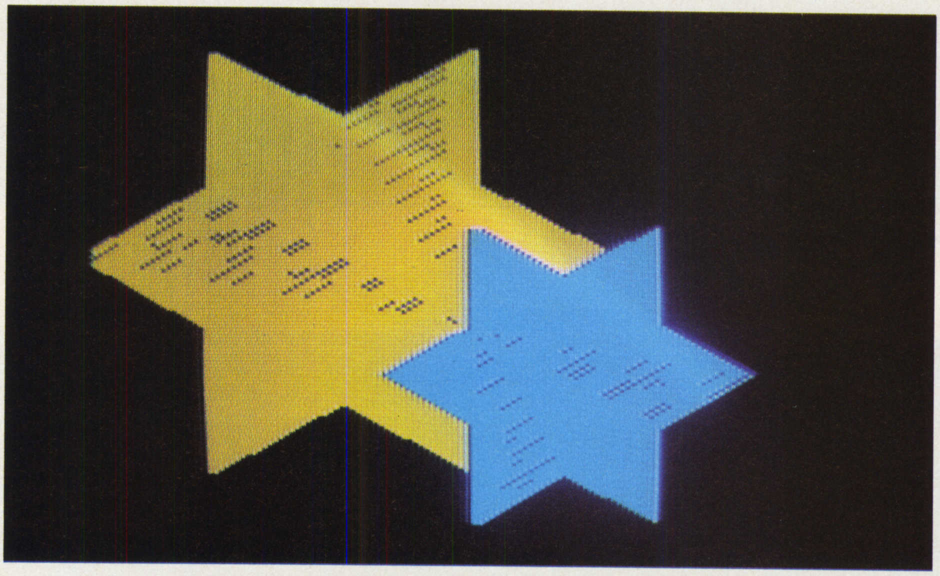
Opposite page:

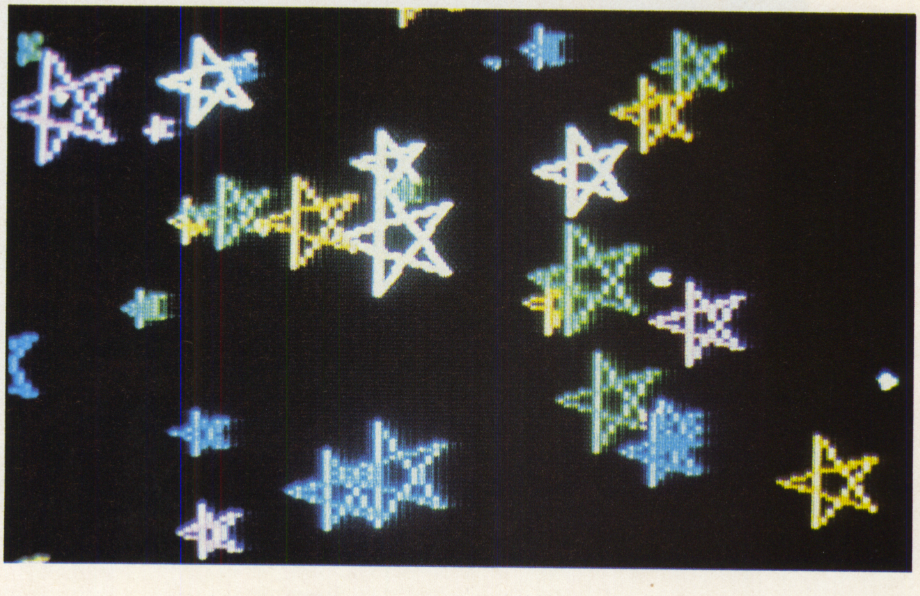
Top: **Big Star** *See procedure on page 74.*

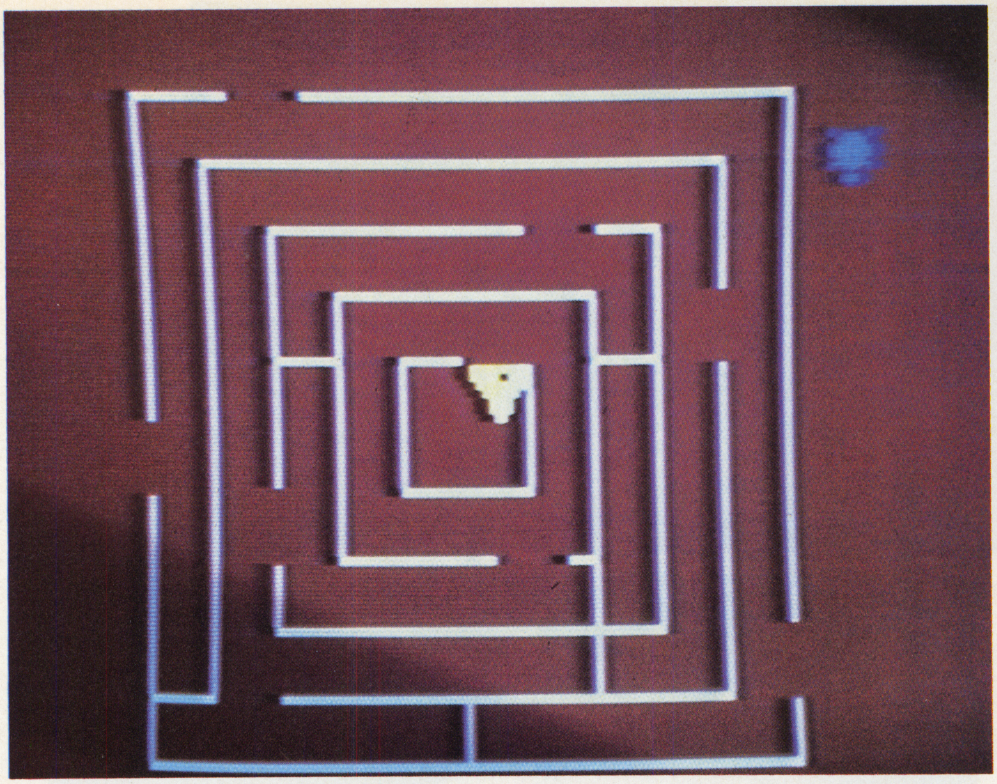
Center: **Big Star** *See procedure on page 74.*

Bottom: **Put Star** *See procedure on page 83.*

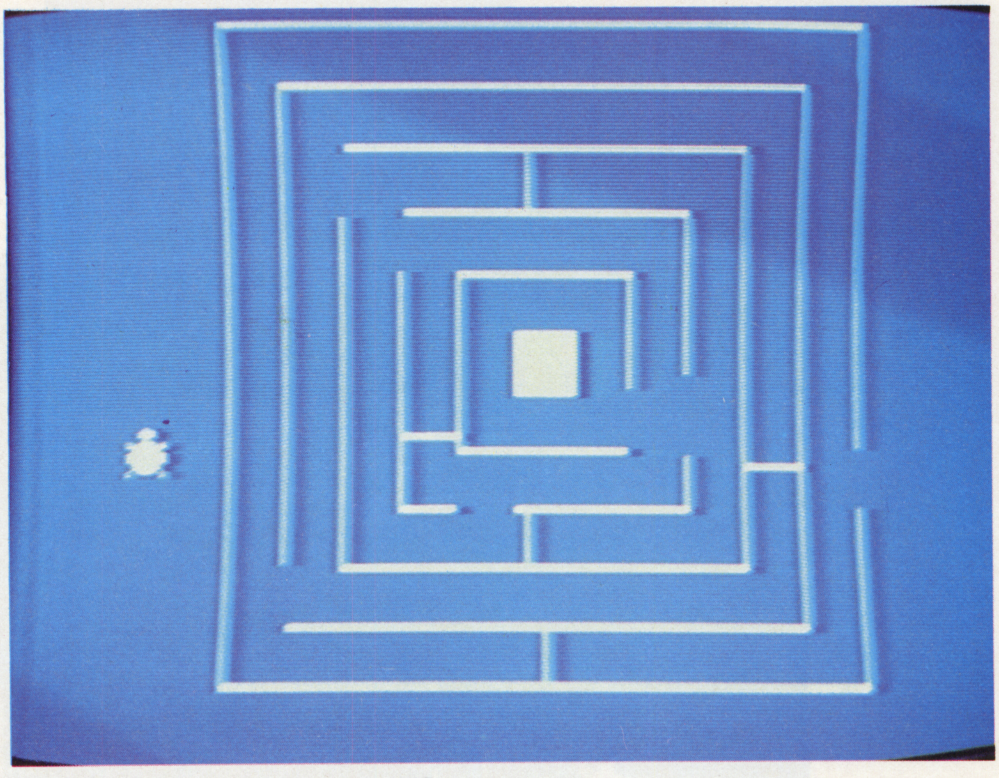








Maze 1 *See procedure on page 109.*



Maze 2 *See procedure on page 110.*

4

Around and Around and Over and Over

CIRCLES AND ARCS

How To Draw a Circle

After having the chance to play with the Turtle for a while and drawing a few shapes, lots of people want to know how to draw a circle. It's easy! Below are a number of different ways to draw circles using Logo.

First, think about what a circle is. Try walking around in a circle as if you were the Turtle. To make a circle you have to take very small steps followed by very small turns. It takes quite a few steps to get all the way around though, doesn't it?

In Logo, a circle is a shape made up of many small sides (or steps) combined with many small turns. The Total Turtle Trip says that in order to get all the way around a shape and end up back where you started, the turns have to add up to 360 degrees. So to draw a circle, you need turns that add up to 360 degrees and enough small sides to form a circle.

How about using REPEAT as a shortcut?

`REPEAT 360 [RT 1 FD 1]`—Makes a circle with 360 one-degree turns (one Total Turtle Trip) and 360 sides, each one turtle step long.

`REPEAT 36 [LT 10 FD 1]`—Makes a circle with 36 ten-degree left turns (36×10=360, or one Total Turtle Trip) and 36 sides, each one turtle step long.

REPEAT 10 [RT 36 FD 2]—Makes a circle with 10 thirty-six-degree right turns (10×36=360, or one Total Turtle Trip) and 10 sides, each two turtle steps long.

Experiment using these formulas for different circles. Is there any difference in what you see on your screen? How would you go about making circles of different sizes? Remember, the number of degrees has to add up to 360, but you could vary the number of turtle steps. Try experimenting with different numbers to see what you get.

How To Draw an Arc

An arc is a curved line, or a part of a circle. Say you wanted to draw a ¼ circle. In that case you would have the Turtle draw a line that was one-fourth the distance around a circle. Remember that in order to go all the way around a circle the degrees of the turns have to add up to 360; if you're drawing an arc that is only part of a circle, the number 360 can help you. If you wanted to draw an arc one-fourth of the way around, for example, the number of REPEATs would equal one-fourth of 360, or 90. An arc that is one-half of a circle would have REPEATs totalling half of 360 or 180. Try some different sized arcs by dividing the number 360:

REPEAT 180 [FD 3 RT 1]—Makes a half circle.
REPEAT 90 [FD 1 RT 1]—Makes a 90-degree arc, or a quarter circle.
REPEAT 45 [FD 1 RT 2]—Also makes a 90-degree arc or a quarter circle. (45 turns×2 degrees=90.)

ARC AND CIRCLE PRIMITIVES

Apple Logo has some arc and circle primitives built in.

Circle Primitives

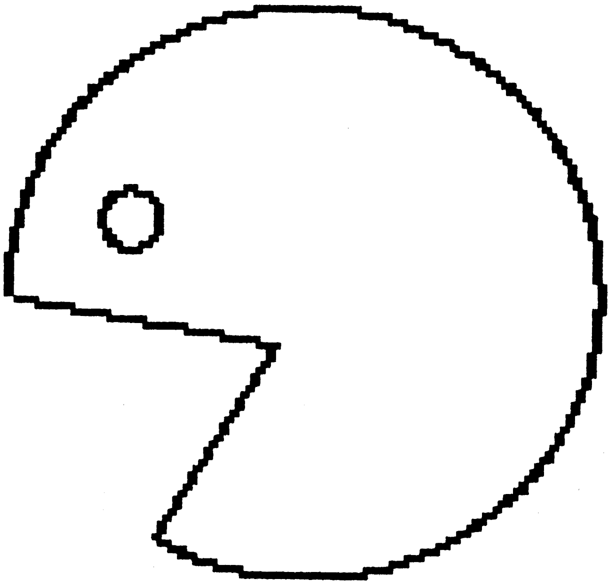
The circle primitives are CIRCLER (for circle right) and CIRCLEL (for circle left). CIRCLER and CIRCLEL are followed by a space and an input—a number that stands for the size of the circle. This number is the radius of the circle, or the distance from the center of the circle to the edge. CIRCLER 10 moves the Turtle to the right to draw a circle with a radius of 10 turtle steps. CIRCLEL 50 sends the Turtle to the left to make a circle with a radius of 50 turtle steps.

Arc Primitives

The arc primitives are ARCR (for arc right) and ARCL (for arc left). Arc primitives need two input numbers. The first number stands for the radius of the arc. The second number stands for the number of degrees in the arc.

ARCR 10 90 makes an arc with a 10-turtle-step radius, 90 degrees to the right. ARCL 90 10 makes an arc with a 90-turtle-step radius 10 degrees to the left. Some interesting effects can be achieved when you vary your input numbers greatly in ARC primitives, to make wavy lines, etc.

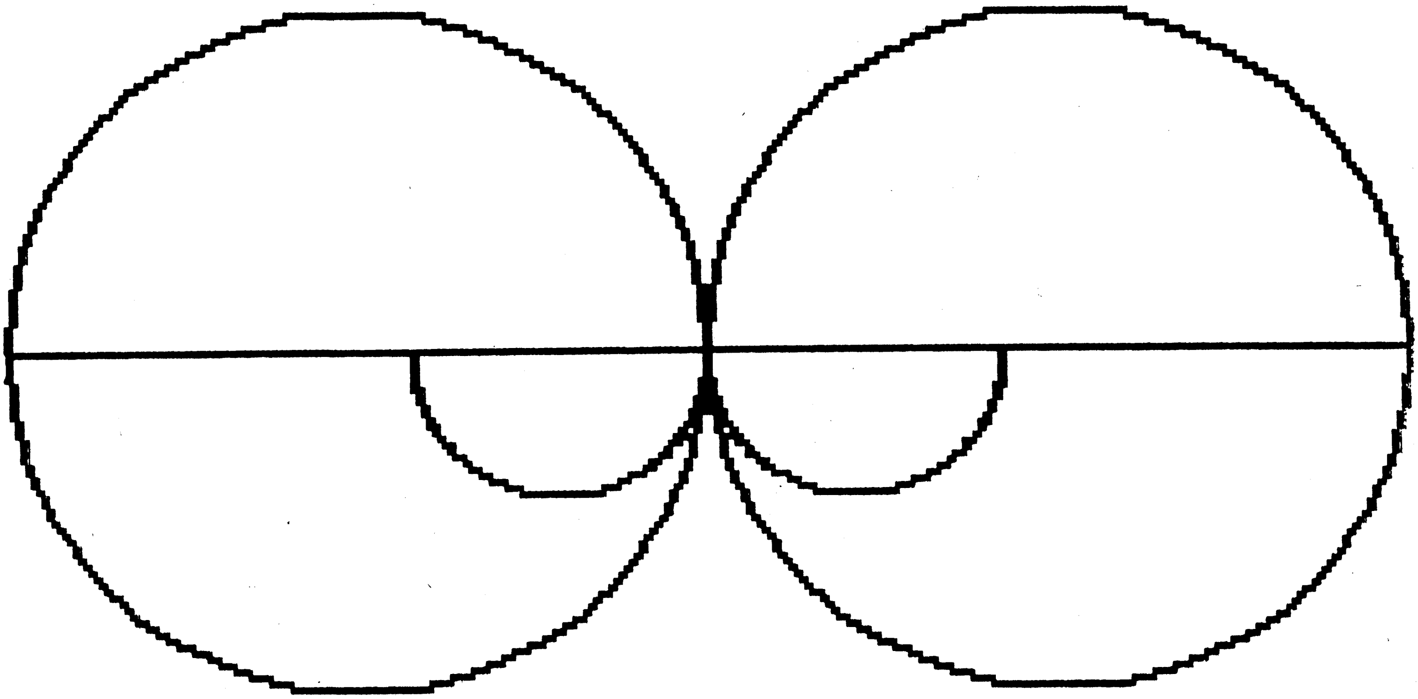
See the following procedures for an idea of what you can do with circles and arcs.



Pacman

(See also color insert, page 62.)

```
TO EYE
REPEAT 36 [FD 1 RT 10]
END

TO BODY
REPEAT 300 [FD 1 RT 1]
RT 90 FD 45
LT 110 FD 50
END

TO PACMAN
BODY
PU RT 120 FD 26 PD
EYE
END
```



Garfield's Eyes

(See also color insert, page 62.)

```
TO G.EYES
HT CIRCLER 68 CIRCLEL 68 RT 90 FD 135
BK 270 FD 135
RT 90
REPEAT 90 [FD 1 RT 2]
PU HOME
PD RT 180
REPEAT 90 [FD 1 LT 2]
END
```

G.EYES uses the **Apple Logo** CIRCLER and CIRCLEL primitives as well as arcs that can be used with any version of Logo.

VARIABLES: THE PARTS THAT CHANGE

So far, the **STAR** procedure we've been using always makes a **STAR** 45 turtle steps long on each side:

```
TO STAR
REPEAT 5 [FD 45 RT 144]
```

Now, we'll write a **STAR** procedure that will make stars of many different sizes. You can change the size of the stars by using a variable. A variable is a part of a procedure that changes when you tell it to. Instead of typing the number 45 after **FD** in the **STAR** procedure, you type in the variable (**:STEPS**):

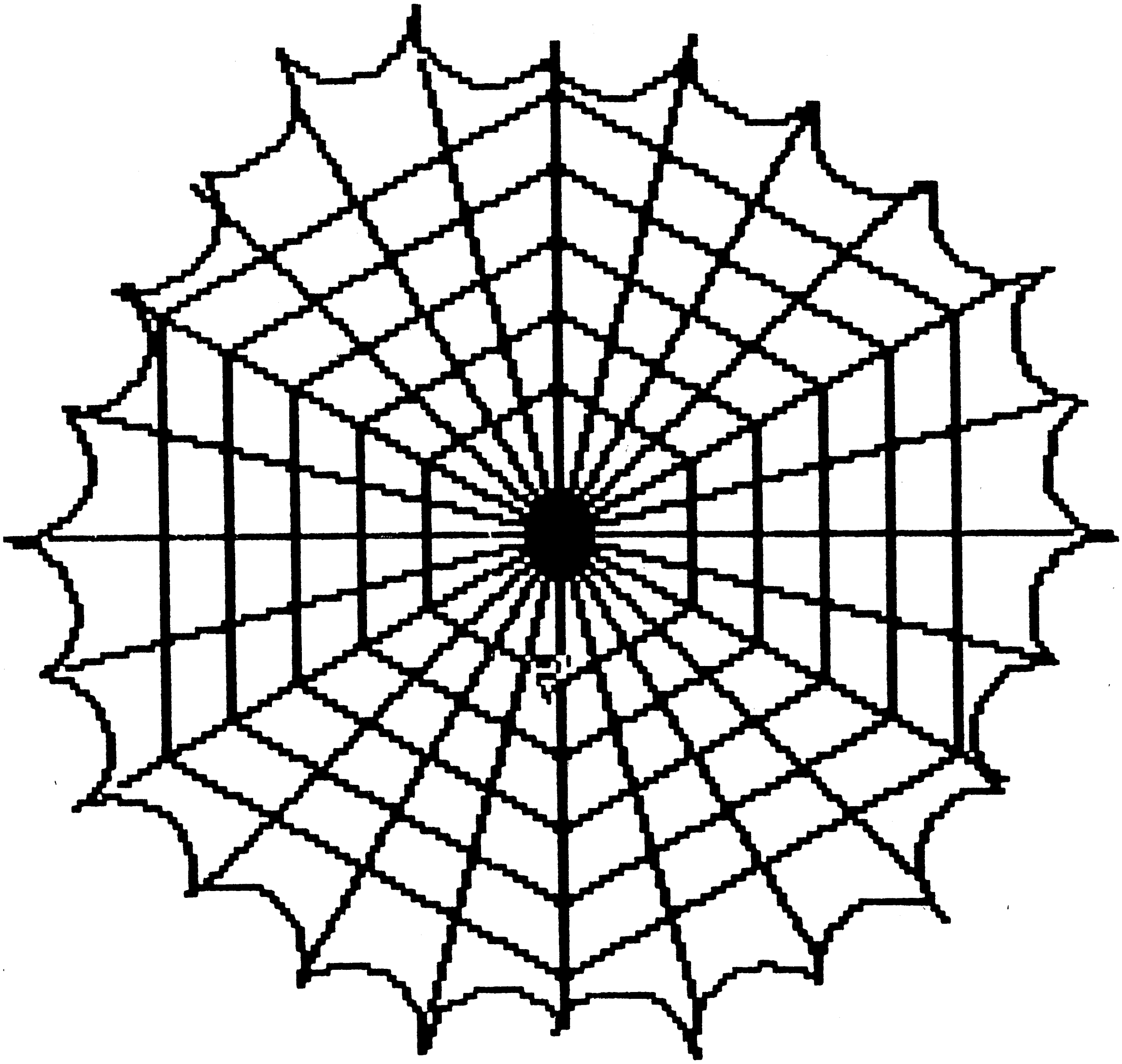
```
TO STAR :STEPS
REPEAT 5 [FD :STEPS RT 144]
END
```

Now to draw a **STAR** on your screen, type **STAR** and a number. The number is the variable, or the number of steps you want the star to be on each side. When you type in the number you are telling the Turtle how many steps to take each time it goes **FD**:

STAR 35 makes a **STAR** 35 turtle steps long on each side.
STAR 65 makes a **STAR** 65 turtle steps long on each side.
STAR 5 makes a **STAR** only 5 turtle steps long on each side.

Variables have names, just like procedures do. The variable in **STAR** is called **STEPS**. Variables always have : (colon) in front of them when you write them into a procedure. You don't use the dots, though, when you're using the procedure.

You can also use more than one variable when writing your procedures. See the examples on the following pages for procedures using variables.



The following program uses **Apple Logo**'s arc primitives and a :SIZE variable to help the Turtle weave a spiderweb. This procedure uses the abbreviation AL for ARCL, or arc left. (See also color insert, page 63.)

```
TO TARANTULA
FULLSCREEN
SPIKES
CURVES
INWEB 90
END

TO SPIKES
REPEAT 24 [FD 100 BK 100 RT 15]
FD 100
END
```

```
TO AL
ARCL 15 140
END

TO CURVES
RT 150 AL RT 180 AL
RT 150 AL RT 144 AL
RT 180 AL RT 144 AL
RT 138 AL RT 180 AL
RT 138 AL RT 138 AL
RT 180 AL RT 180 AL
RT 138 AL RT 138 AL
RT 138 AL RT 180 AL
RT 150 AL RT 180 AL
RT 138 AL RT 138 AL
RT 165 AL RT 138 AL
RT 180 AL
RT 180 AL

TO INWEB :SIZE
IF :SIZE<30 [STOP]
PU HOME FD :SIZE RT 120 PD
REPEAT 6 [FD :SIZE RT 60]
INWEB :SIZE - 15
END
```

RECURSION

In Logo, the procedures you write can use themselves as subprocedures to repeat themselves over and over again. This way of repeating is called *recursion.* We can use the STARDESIGN procedure below to get an idea of how recursion works:

```
TO STARDESIGN2
STAR
RT 36
STARDESIGN 2
END
```

This procedure goes on drawing STARDESIGN2 over and over again. Here's how it works: The first line of STARDESIGN2 tells the Turtle to draw a STAR. Next it turns right 36 degrees. Then, in the third line of the program it comes to its own name. When it sees its own name, it goes back to the first line of the program and draws another STAR, turns right 36 degrees and sees its own name again! This recursive procedure will continue until you press a STOP or BREAK key.

Recursive procedures can be used with variables and arithmetic to make them more versatile. BIGSTAR uses a variable called :SIZE. The procedure begins with the Turtle drawing the outer star at whatever size variable you tell it to. The next line of the procedure tells the Turtle to draw the star again, but at :SIZE – 1. The second star the Turtle draws will be one Turtle step smaller on each side than the first one. Because this procedure is recursive, it will go around and around again until you tell it to stop by pressing a STOP or BREAK key. But BIGSTAR has a surprise in it, too!

What do you suppose happens when the :SIZE – 1 variable in BIGSTAR gets to a point where the sides of the star become less than 0? Try it and see! You can also look at page 65 in the color insert.

```
TO BIGSTAR :SIZE
REPEAT 5 [FD :SIZE RT 144 FD :SIZE LT 72]
BIGSTAR :SIZE  – 1
END
```

IF

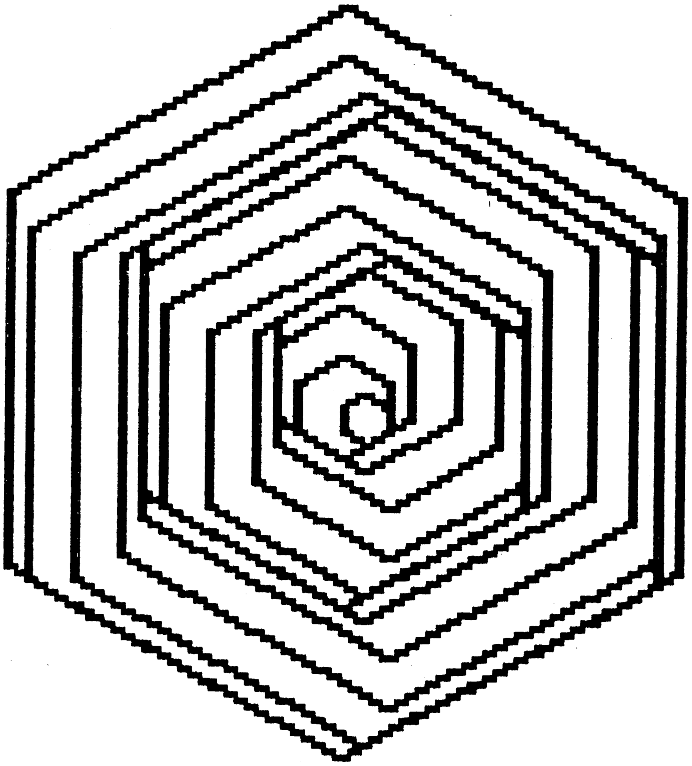
If is the magic word to use if you want to stop a recursive procedure automatically. You simply add a line to your recursive procedure that tells the Turtle what to do IF certain conditions occur. For instance, you could stop BIGSTAR by adding the condition:

```
IF :SIZE< 0 STOP
```

With the above command added to BIGSTAR, the stars would not grow out the other side when the variable becomes less than 0.

HEX7 :STEPS is another example of a recursive procedure that contains an IF statement in it.

IF statements are called *conditionals*, and they can do more than just stop recursive programs. Suppose we wanted the IF statement in BIGSTAR to do something besides make it stop? You can use the IF statement to tell the Turtle to do anything it knows how to do. For **Apple** and **Atari Logo**, whatever you want the Turtle to do must be enclosed in brackets ([ ]). Let's use an IF statement to combine more than one procedure:



```
TO HEX7 :STEPS
IF :STEPS<10 [STOP]
REPEAT 7 [FD :STEPS LT 60]
HEX7 :STEPS - 5
END
```

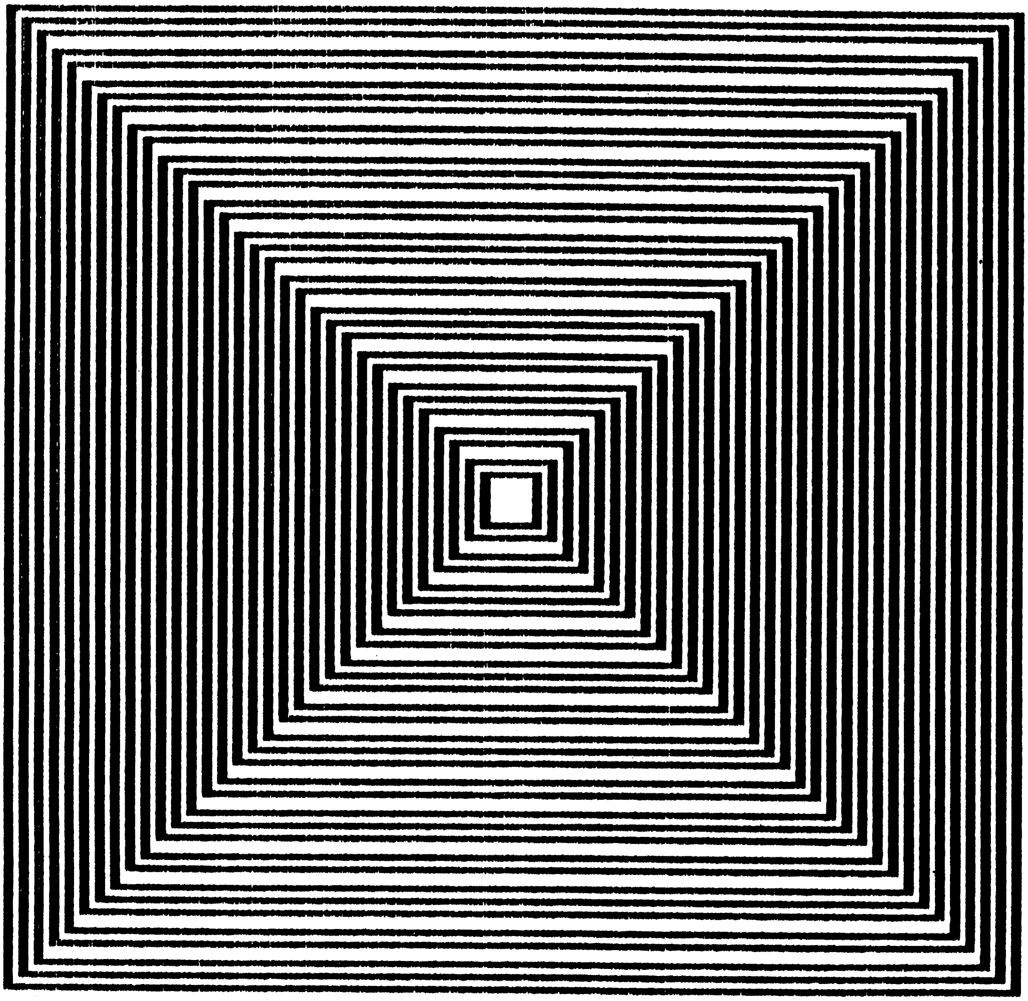
```
TO STARHEX :SIZE
IF :SIZE<20 [WAIT 300 CS HEX7 60]
REPEAT 5 [FD :SIZE RT 144]
PU FD 10 PD
STARHEX :SIZE -5
END
```

STARHEX plus an input number for :SIZE tells the Turtle to draw STARHEX until the size of the star is less than 20; wait five seconds, then clear the screen and draw the recursive HEX7 starting with a :STEPS variable of 60 turtle steps on each side.

Recursion can also be used with more than one variable and IF statements to make some very interesting and versatile procedures.

PYRAMID first makes the smallest size SQUARE. Next, it positions the Turtle by moving it over and back a certain distance, :DIST. Then, it adds the :DIST to the :SIZE twice (:SIZE + :DIST*2).* The procedure goes back and repeats this process until (IF) the size of the largest SQUARE is 200 turtle steps long on each side.

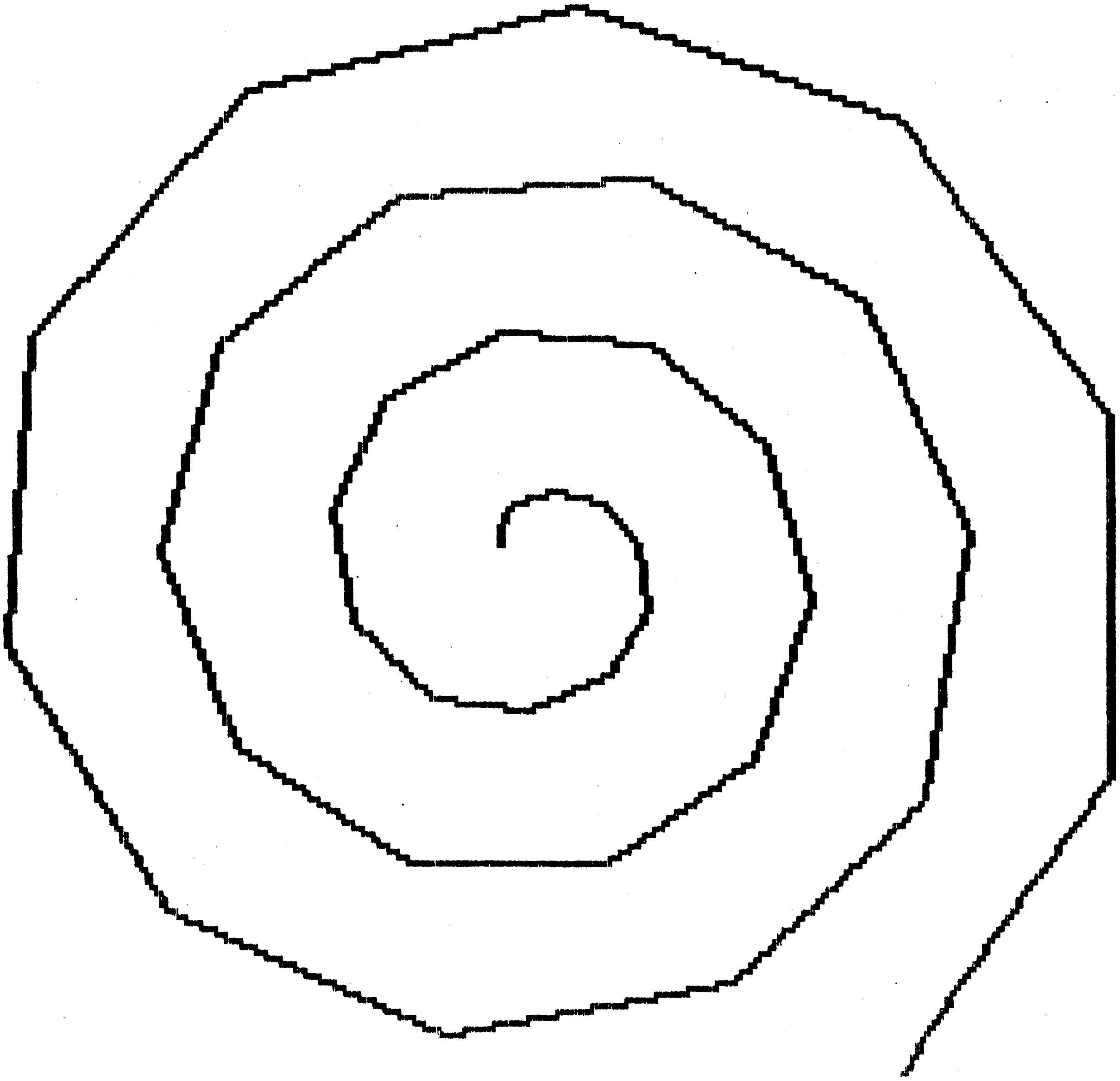
*NOTE: When performing arithmetic: + is the computer character used for addition; – for subtraction; * for multiplication; / for division; > for “greater than”; < for “less than”; = for equals.



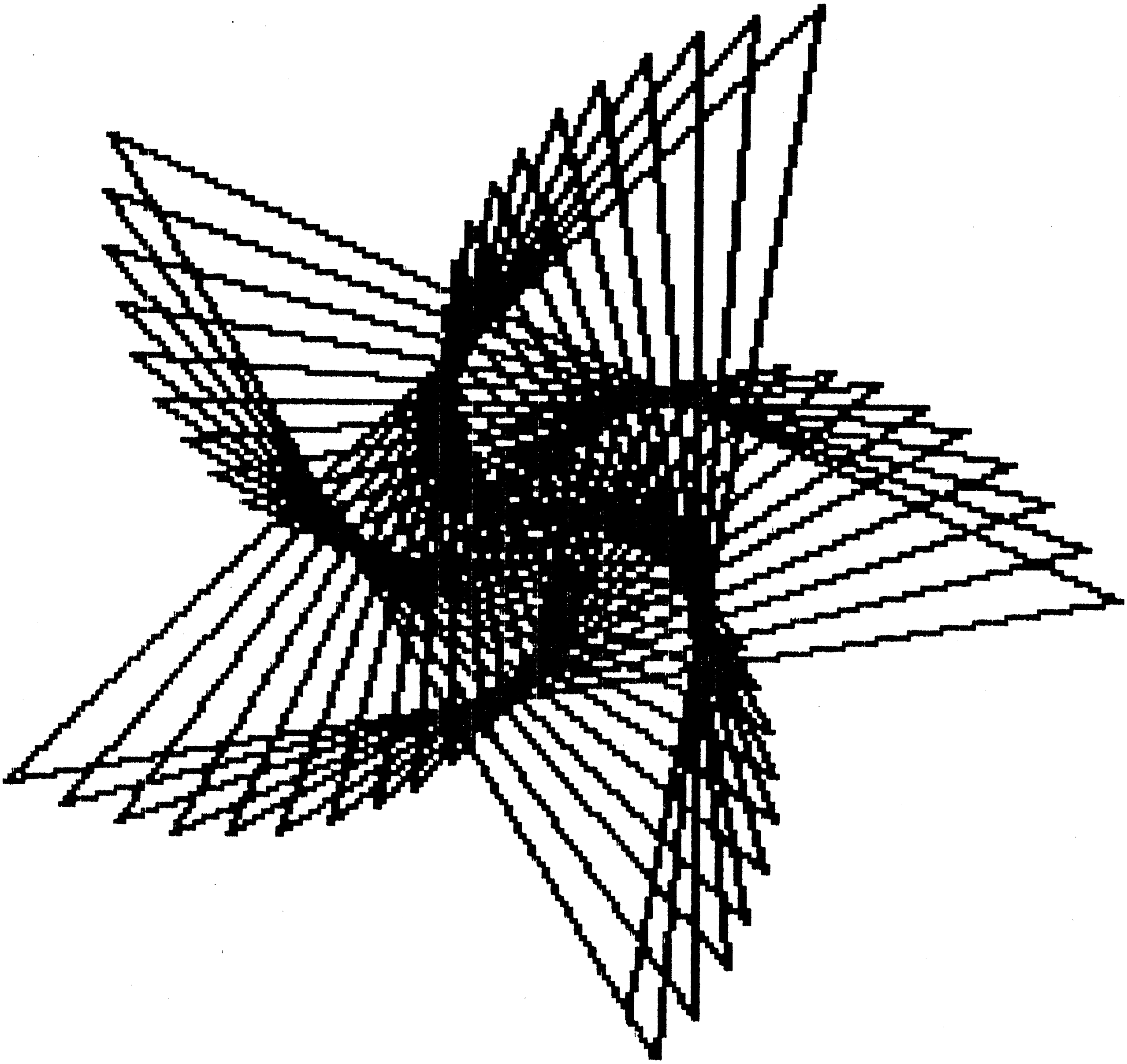
```
TO PYRAMID :SIZE :DIS
IF :SIZE>200 [STOP]
SQUARE :SIZE
PU LT 90 FD :DIS
RT 90 BK :DIS PD
PYRAMID :SIZE+  :DIS*2 :DIS
END

TO SQUARE :SIZE
REPEAT 4 [FD :SIZE RT 90]
END
```

The following procedures show how recursion and variables work together (see also color insert, page 64):

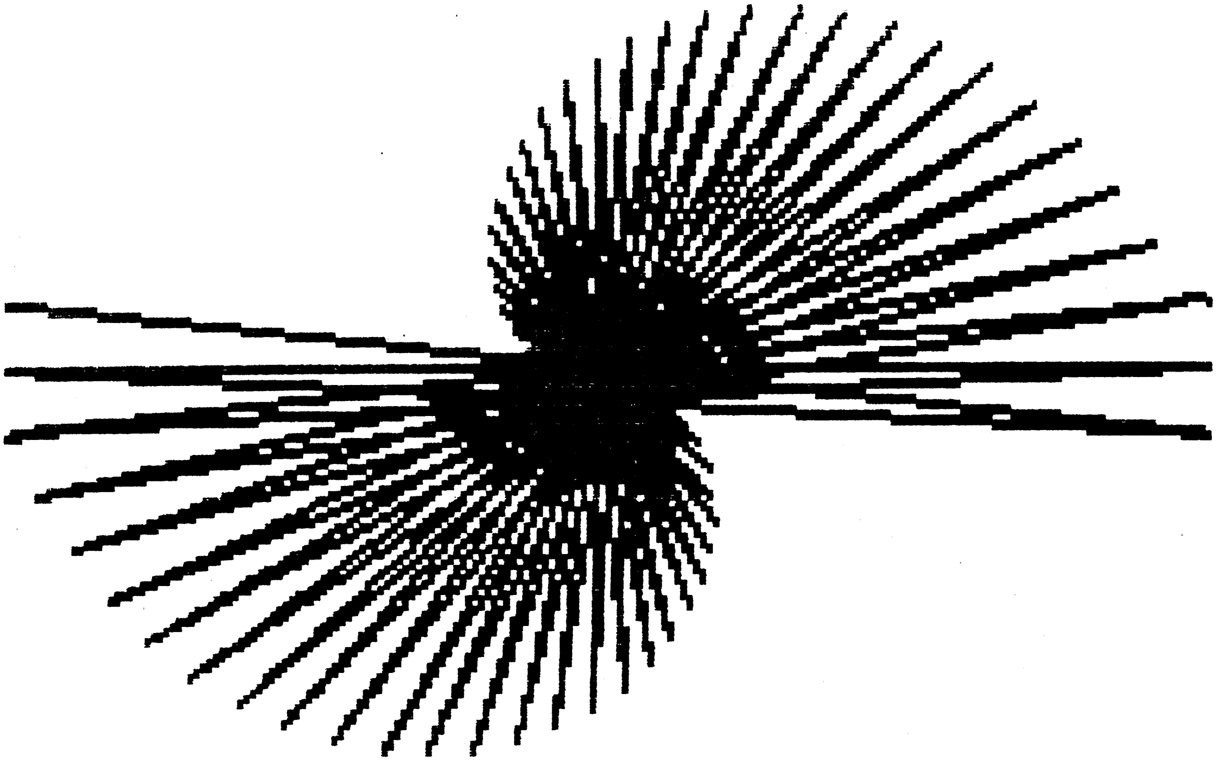
```
TO SPIRAL :SIZE :TURN :GROW
IF :SIZE > 200 [STOP]
FD :SIZE RT :TURN
SPIRAL :SIZE +  :GROW :TURN :GROW
END
```



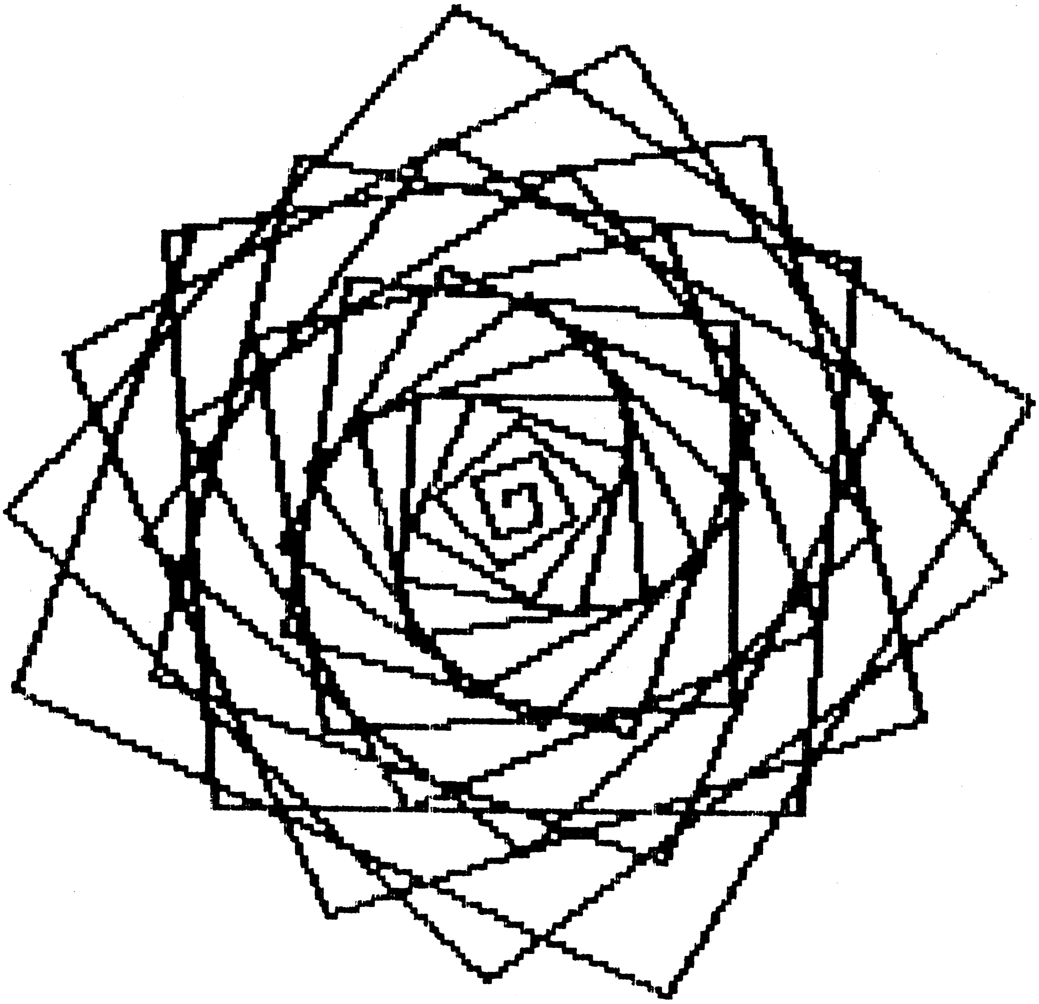
SPIRAL 3 34 3



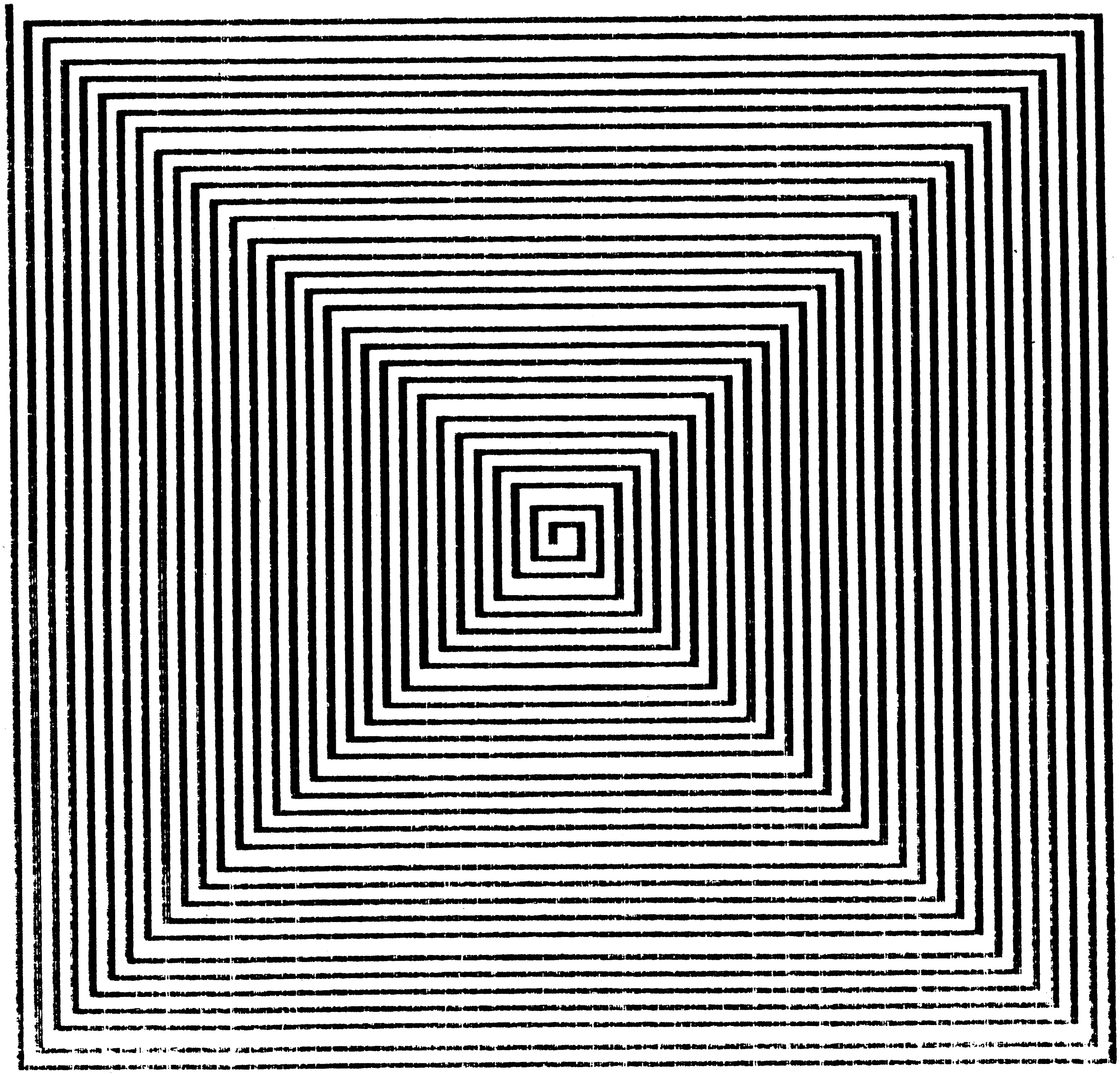
SPIRAL 5 90 3



SPIRAL 3 145 3



SPIRAL 5 183 5



SPIRAL 5 96 4

RANDOM

Sometimes in your Logo programs you may not want to know exactly how they will turn out. Or, you may want to let your computer do some of the work for you. Then it's time to use the command RANDOM. (RANDOM is a primitive that takes one number as input.) If you type FD RANDOM 99 the Turtle will go FD a RANDOM amount between 0 and 98 turtle steps, depending on which of those numbers the computer chooses. RT RANDOM 45 turns the Turtle RT a RANDOM amount between 0 and 45 degrees. RANDOM always chooses a number between 0 and one less than the number you give it as input. FD 1 + RANDOM 90 moves the Turtle FD a RANDOM amount between 0 and 90 and adds one to that amount. LT 5 + RANDOM 120 turns the Turtle a RANDOM amount between 0 and 120 degrees and adds five to the turn.

You can use RANDOM in procedures that have variables. To make five STARs of RANDOM sizes (using the STAR :SIZE procedure) you would type:

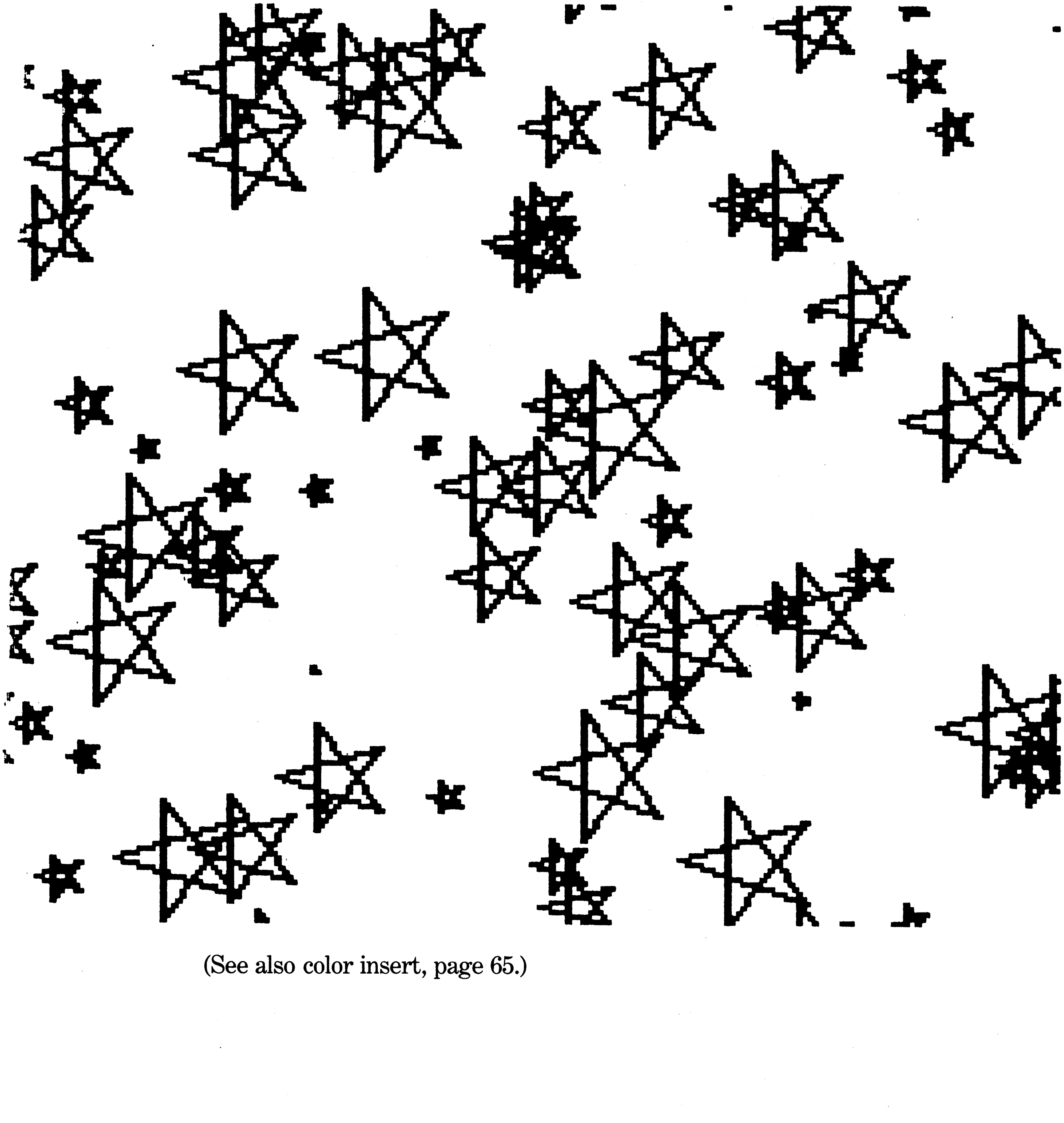
```
REPEAT 5 [STAR 5 + RANDOM 50 PU FD 20 PD]
```

The Turtle will draw five STARS of various sizes between five and 54 turtle steps on each side, and will move 20 turtle steps FD between each STAR.

The important thing to remember about RANDOM is that the computer chooses a different number each time you use a procedure, so the results are not always the same. The three procedures that follow will put STARS of RANDOM sizes and colors at RANDOM positions on your screen. The word LIST in PUTSTAR is used instead of the brackets for setting RANDOM positions with the SETPOS command in **Apple** and **Atari Logo.** This procedure was written in **Apple Logo**; the RANDOM numbers vary for different versions of this procedure. **TI Logo** can choose RANDOM numbers between 0 and 9 only.

```
TO STAR :SIZE
REPEAT 5 [FD :SIZE RT 144]
END

TO PUTSTAR
PU SETPOS LIST RANDOM 280 RANDOM 240
PD STAR 5 + RANDOM 35
END

TO PUTSTARS :HOWMANY?
FULLSCREEN
HT
REPEAT :HOWMANY? [SETPC 1 + RANDOM 5 PUTSTAR]
END
```

:HOWMANY? is the variable that allows you to choose how many STARS you want on your screen.



(See also color insert, page 65.)

DOT

DOT is a primitive in **Apple Logo** used to make dots on your screen. DOT takes two numbers as inputs. The first is an *x* coordinate number and the second a *y* coordinate number. (See p. 87 for *x* and *y* coordinates.) The input numbers for the DOT command are always put in brackets. DOT [0 0] places a dot in the center of your screen. To achieve special effects with the DOT command like we did with the SKY procedure in BLASTOFF on page 107, we wrote DOT into a procedure:

```
TO DOTS
DOT LIST RANDOM 280 RANDOM 240
END
```

This procedure places one DOT at random *x* and *y* positions on the screen.

BUGBUSTER: The word LIST is used with RANDOM commands instead of brackets when it's used with the SETPOS command in **Apple** and **Atari Logo.**

For similar effects with other versions of Logo, choose one of the procedures below:

TI LOGO:

```
TO SKY
TELL TURTLE
HT CB 1
REPEAT 200 [PU RT 45 FD RANDOM*7 FD 1]
END
```

ATARI or TERRAPIN/KRELL LOGO:

```
TO SKY
FULLSCREEN (or FS)
HT SETBG 0
REPEAT 250 [PU RT RANDOM 5 RANDOM 20*5 PD FD 1]
END
```

WORDS TO REMEMBER: CHAPTER FOUR

ARC—An arc is a curved line, or part of a circle. Arcs are drawn in Logo by using the same method used for drawing circles. **Apple Logo** also has arc and circle primitives.

RADIUS—The distance from the center of a circle to its edge.

VARIABLE—The part or parts of a Logo procedure that change. Variables must be entered into a procedure with a colon (:) in front of them.

RECURSION—When a procedure calls or repeats itself as part of the procedure, it is called recursion.

IF—A primitive used to instruct a procedure. IF, followed by a list of instructions, tells Logo what to do if certain conditions should occur. The IF command is also known as a conditional in Logo.

RANDOM—A Logo primitive that allows the computer to choose what will happen. RANDOM takes one number as input.

DOT—A primitive in **Apple Logo** that allows you to put dots on your screen.

5

Turtles and More Turtles

OTHER WAYS TO MOVE THE TURTLE

There's more than one way to move a Turtle around your screen. Besides the commands you already know—FD, BK, RT, and LT—each version of Logo has a few other commands that move and position the Turtle.

Setting the Turtle's Position Using X and Y

You can move the Turtle using what are called *x* and *y* positions or *coordinates.* These coordinates are determined by measuring the distance from the center of the screen or HOME. Imagine your computer screen filled with horizontal and vertical lines. The horizontal lines measure the x coordinates and the vertical lines are the y coordinates. Both x and y coordinates are 0 when the Turtle is HOME. X and y coordinates can have both positive and negative values. Positive x positions are to the right of 0. Negative x positions are to the left of 0. Positive y positions are above 0, negative y positions are below 0.

To set the Turtle's position using x and y, you must first enter the correct command from the list below, followed by 1 or 2 input numbers. The first number stands for the x coordinate and the second input number stands for the y coordinate. Do some experimenting to find out more about how x and y coordinates work.

BUGBUSTER—Be sure to enter negative numbers carefully when using SET commands. Space the minus sign carefully or Logo may think you are trying to subtract. There is no space between the minus sign and the number when inputting a negative number.

TABLE OF SET COMMANDS FOR SETTING TURTLE POSITIONS

PRIMITIVE	INPUTS NEEDED	FUNCTION	VERSION
SETX	1 NUMBER	SETS TURTLE'S X POSITION	APPLE, ATARI, TERRAPIN/KRELL
SETY	1 NUMBER	SETS TURTLE'S Y POSITION	APPLE, ATARI, TERRAPIN/KRELL
SETPOS	2 NUMBERS IN BRACKETS	SETS TURTLE'S POSITION USING X AND Y COORDINATES	APPLE, ATARI
SX	1 NUMBER	SETS TURTLE'S X POSITION	TEXAS INSTRUMENTS
SY	1 NUMBER	SETS TURTLE'S Y POSITION (negative y coordinates must be in parentheses.)	TEXAS INSTRUMENTS
SXY	2 NUMBERS	SETS TURTLE'S POSITION USING X AND Y COORDINATES (negative x coordinates must be in parentheses.)	TEXAS INSTRUMENTS
SETXY	2 NUMBERS	SETS TURTLE'S POSITION USING BOTH X AND Y COORDINATES	TERRAPIN/KRELL

Turning the Turtle: Setting the Heading

The direction in which the Turtle points is called its heading. The Turtle's heading is based on the familiar Total Turtle Trip of 360 degrees. When the Turtle is pointing toward the top of the screen, its heading is 0 (0 degrees).

When the Turtle is pointing directly to the right, its heading is 90 (90 degrees). Straight down is 180.

You can set the Turtle's heading using the correct commands for your system and an input number. Setting the Turtle's heading is a more exact method of turning the Turtle than RT and LT commands, because there's less guesswork involved. You know right away what numbers you have to choose from.

A heading of 360 means that the Turtle is once again pointing straight up. Can you guess why that is? Think back to the Total Turtle Trip and what it means, or read over the section on the Total Turtle Trip in Chapter One.

Turtle headings can also be negative numbers. SETH −45 is the same as SETH 315. SETH −90 is the same as SETH 270.

The commands for setting headings are below. They must be followed by an input number:

APPLE: SETH

ATARI: SETH

TI: SETHEADING or SH. In this version of Logo, SETHEADING can also be used with the names of the four directions— :NORTH, :SOUTH, :EAST, and :WEST. Words that follow the SH command in **TI Logo** must have a colon (:) directly in front of them.

TERRAPIN/KRELL: SETHEADING or SETH

TOWARDS

In **Apple** and **Terrapin/Krell Logo,** the TOWARDS command can be used with the SETHEADING or SH commands to point the Turtle towards a certain x y coordinate on the screen.

For example: SH or SETH TOWARDS [45 53] points the Turtle TOWARDS the x coordinate point 45 and the y coordinate point 53. Remember when using the TOWARDS commands to put the two input numbers in brackets.

TOWARDS can also be used to find out what heading is needed to point the Turtle TOWARDS a certain x y coordinate on the screen. To do this, you first type PRINT TOWARDS followed by the two coordinates in brackets. Press RETURN, and the heading you need will be printed out on the screen.

THE FOUR TURTLES OF ATARI

There are four Turtles in **Atari Logo** (and they look like Turtles!). Each of the Turtles has a turtle number assigned to it:

Turtle 0
Turtle 1
Turtle 2
Turtle 3

In order to give commands to an **Atari Turtle**, you must talk directly to the Turtle you want to use. You do this by using the primitive commands TELL and ASK. Both these commands need to be followed by a turtle number or numbers as inputs to specify which Turtle you're talking to:

TELL

```
TELL 0
FD 45
```

will make Turtle 0 go forward 45 turtle steps.

```
TELL 3
PU BK 43 PD
```

will make Turtle 3 pick up its pen, go back 43 turtle steps, and put its pen down again.

Once you've typed the command TELL and a turtle number, you don't need to type TELL again until you want to talk to another Turtle.

You can also talk to more than one Turtle at a time. TELL [0 1 2 3] means that you are talking to all four Turtles. You can address any combination of Turtles you want. Just make sure to enclose the turtle numbers in brackets if you're talking to more than one of them at a time.

ASK

Suppose you want to talk to more than one Turtle, but you want each one to do something different? Then it's time to use the ASK command.

Try the following example:

```
TELL [0 1 2 3]
PU HOME ST
```

This gets all four Turtles home and showing themselves. You will only be able to see one Turtle on your screen though, because the other three Turtles are hiding behind it.

Then type the following:

```
ASK 0 [RT 45]
ASK 1 [LT 45]
ASK 2 [RT 135]
ASK 3 [LT 135]
FD 50
```

What happens? Each Turtle goes forward 50 turtle steps in a different direction. Notice in the above example that all commands following the ASK command must be enclosed in brackets, otherwise they will not output.

You can also ASK more than one Turtle at a time to do something. Enclose the turtle numbers in brackets following the ASK command.

TELL and ASK commands can also be written as part of your procedures in **Atari Logo.** In a procedure, the above example would be written like this:

```
TO X
TELL [0 1 2 3]
PU HOME PD ST
ASK 0 [RT 45]
ASK 1 [LT 45]
ASK 2 [RT 135]
ASK 3 [LT 135]
FD 50
END
```

WHO

WHO is a primitive in **Atari Logo** that carries the numbers of the currently active Turtles. If you type PR WHO, **Atari Logo** will respond with the numbers of the active Turtles.

WHO can also be used with arithmetic. EACH [FD 10*WHO] (see EACH, next section) will cause each Turtle to move FORWARD 10 times its own number. That is, Turtle 1 would take 10 turtle steps forward because 10×1=10. Turtle 2 would take 20 steps, and so on. Try the WHO command with some other simple combinations.

EACH

If you're talking to more than one Turtle, you can have EACH of them obey a command separately with the primitive EACH. EACH is always followed by brackets containing a list of instructions for EACH Turtle to carry out. The procedure below uses both the WHO and EACH commands and a nested procedure called CIRCLE.

```
TO CIRCLE
REPEAT 360 [FD 1 RT 1]
END

TO T.FLOWER
TELL [0 1 2 3]
EACH [SETH 90*WHO]
CIRCLE
EACH [SETH 45 + 90*WHO]
CIRCLE
END
```

DYNAMIC TURTLES

You can use **Atari** Turtles for animation. Try this:

```
TELL 0
SETSP 55 (set speed)
```

Turtle 0 will now be moving across your screen at a speed of 55.

When a Turtle is set for speed, it is said to be *dynamic*. Dynamic Turtles can move with their pens up or down, depending on the effects you want to create. They can be given HEADING, POSITION, and COLOR commands, and any of these commands can be changed while the Turtle is in motion.

SPEED for Dynamic Turtles can be set at –127 to 127 turtle steps per second. Negative numbers move the Turtle backwards. Try the program below for FLYING TURTLES:

```
TO FLY
TELL [0 1 2 3]
PU
EACH [SETH 90*WHO SETSP 55]
WAIT 120
EACH [SETH -90*WHO SETSP -55*WHO] WAIT 120
FLY
END
```

You will have to press BREAK to stop this recursive procedure.

COLLISION DETECTION

Atari Logo can detect when a Turtle hits or collides with a line, or when a Turtle collides with another Turtle. This is called *collision detection.* Collisions are coded by number. For example, when Turtle 0 collides with a line drawn by PEN 0, the code number is 0. There is a table of collision code numbers included on page 93.

Using the collision code numbers with the primitive WHEN lets you set up your own collisions. When you enter a WHEN command in **Atari Logo**, it calls up a WHEN "demon." This demon has only one job—to watch for collisions. Then, when the collisions take place, it gives your collision instructions to the Turtle. WHEN is always followed by a collision code number and a list of instructions inside brackets. For example:

```
WHEN 8 [ASK 2 RT 45 FD 50]
```

gives the instructions that WHEN collision number 8 takes place, Turtle 2 should turn RIGHT 45 degrees, and go FORWARD 50.

BUGBUSTER—If you have a WHEN demon watching for collisions for one Turtle, and you start talking to a different Turtle, the new Turtle will be the one to obey the WHEN commands. To separate Turtles and WHEN demons, use the ASK command (see page 90). To stop a WHEN demon from watching, type CS, or type WHEN followed by your collision number and [ ]. For example:

```
WHEN 0 [ ].
```

TABLE OF COLLISION CODES (ATARI LOGO ONLY)

CODE NUMBER		INPUTS		DESCRIPTION
COLLISION	SPECIAL EVENT	TURTLE/PEN NUMBER		
0		0	0	
1		0	1	
2		0	2	
	3			button on joystick is pressed
4		1	0	
5		1	1	
6		1	2	
	7			once per second
8		2	0	
9		2	1	
10		2	2	
11	Not Used			
12		3	0	
13		3	1	
14		3	2	
	15			joystick position is changed
16		3	0	
17		3	1	
18		3	2	
19		0	1	
20		0	2	
21		1	2	

From *Atari Logo Guide* ©1983 Logo Computer Systems, Inc.

Try some of the following examples of collision detection to find out more about how it works.

TO BUMP is a procedure that uses all possible examples of collisions between Turtles:

```
TO BUMP
TELL [0 1 2 3]
ST HOME FS PD
WHEN 16 [ASK [3 0] [RT 180]]
WHEN 17 [ASK [3 1] [RT 180]]
WHEN 18 [ASK [3 2] [RT 180]]
WHEN 19 [ASK [0 1] [RT 180]]
WHEN 20 [REPEAT 10 [SETBG RANDOM 127]]
WHEN 21 [REPEAT 10 [SETC RANDOM 127]]
EACH [SETH 5 + 35*WHO]
SETSP 95
END
```

To INSQUARE is a procedure that shows what happens in Turtle and pen collisions:

```
TO SQUARE :SIZE
REPEAT 4 [FD :SIZE RT 90]
END
```

```
TO INSQUARE
TELL [0 2 3] HT
TELL 1
ST HOME PD
SETPN 1
SETPC 1 7
SQUARE 55
RT 45 PU SETSP 35
WHEN 5 [RT 175]
END
```

When Turtle 1 hits the sides of the SQUARE drawn with PN 1 it will turn RIGHT 175 degrees. This is collision #5 (WHEN 5).

For examples of how collision detection can be used in games, see the MAZE programs on page 109.

SHAPES AND SPRITES

Another way to create animation in Logo is through the use of shapes and sprites. Both **Atari Logo** and **TI Logo** include special ways to make different shapes and use them in your programs.

ATARI SHAPES

In **Atari Logo**, you can change the SHAPE of the Turtles using the SHAPE Editor. To get into the SHAPE Editor, type:

```
EDSH 1 (edit Shape 1)
```

You should now see a grid eight squares wide and 16 squares long on your screen. You can use this grid to make up a SHAPE. Here's how:

Use the CTRL key with the four arrow keys to move the cursor around. If you want to fill in a square on the grid, move the cursor to that square and press the space bar. To erase a filled-in square, move the cursor to that square and press the space bar again. When your SHAPE is finished, press the ESC key.

To have your SHAPE appear on the screen, you must TELL one or more of the Turtles to change into that SHAPE:

```
TELL 0
SETSH 1 (set Shape 1)
SETC 12
HOME
```

Once the Turtle has taken your SHAPE, you can give it SPEED, POSITION, COLOR, etc. Each SHAPE needs a different SHAPE number. Because **Atari** has four Turtles, you can have up to four SHAPES appear on your screen at once. To change a SHAPE back into a Turtle, TELL the Turtle carrying that SHAPE to SETSH 0.

Saving Shapes

You can make up and SAVE an unlimited number of SHAPES in **Atari Logo**, but your computer will only be able to store 16 SHAPES at once in its memory. The rest will have to be SAVED to a cassette or disk.

To SAVE a SHAPE we use the GETSH (get shape) operation. First, give your SHAPE a name. For example, let's say we had a SHAPE named FISH. Type:

MAKE "FISH GETSH 1 (Be sure to give your SHAPE the SHAPE number that belongs to it.)

Now, when you save your work, the SHAPES you've named in this way will be saved along with it. Next time, when you reload your work back into your computer, you will be able to see your shapes again by using the PUTSH (put shape) command:

PUTSH 1 :FISH puts the FISH back into Shape 1, so you can then SETSH with a Turtle.

TEXAS INSTRUMENTS SPRITES

You can create animation in **TI Logo** using Sprites. Sprites are Shape carriers. They are invisible until you give them a shape. **TI Logo** comes with five existing Shapes:

Shape	*Design*
PLANE	
TRUCK	
ROCKET	
BALL	●
BOX	■

There are also 27 blank spaces for you to create Shapes that you make up yourself. To do this, type the command:

MAKESHAPE 7 (ENTER) or MS 7

Your screen will turn green and a grid 16 squares wide and 16 squares high should appear. To make a new Shape in **TI Logo**, you can use any Shape number from 7 through 25. The cursor will be in the upper left hand corner. You move the cursor around the screen with the arrow keys (E S D X). You fill in the squares on the grid using the FCTN key in combination with the arrow keys. If you want to erase your design and start over, press FCTN and CLEAR. When you have finished making a new shape, press FCTN and BACK.

To have your Shape appear on the screen, you need to first tell a Sprite to carry the Shape and give it a position and color. Let's try this with one of the existing Shapes, the rocket ship:

```
TELL 1 (TELL Sprite 1)
CARRY 3 (The code number of the rocket ship)
SC 14 (SETCOLOR 14)
HOME
SH 0 (Set heading toward the top)
SS 55 (Set speed at 55 turtle steps per minute)
```

Now you should have a rocket ship heading up to the top of your screen, wrapping around, and reappearing from the bottom. You can change the color, heading and speed of a Sprite. You can also set x and y coordinates to change the position of a Sprite. If you like, you can even tell all 32 Sprites to carry one Shape (TELL :ALL CARRY 3). Remember, if you want to get the Turtle back, type TELL TURTLE.

Sprites can also be written into procedures. Here's a simple example using the ball shape:

```
TO BOUNCE
TELL 2

CARRY :BALL (You can use Sprite names instead of numbers if you
want. Just remember to use a colon (:) in front of the shape name.)

SC 5
SH 90
SY 0
SS 65
WAIT 30 SH 45 WAIT 30 SH 135 BOUNCE
END
```

This is a recursive procedure. To make the ball disappear, type TELL 2 SC 0. Press FCTN and BACK to stop BOUNCE.

There are other commands to use with Sprites, too:

FREEZE—Stops all the Sprites on your screen.
THAW—Puts the Sprites in motion once again.
BIG—Makes all Shapes 32 squares wide and 32 squares high. (**TI Logo II** only)
SMALL—Makes all Shapes 16 squares wide and 16 squares high. (**TI Logo II** only)

You can talk to more than one Sprite at once [Note: TELL [1 2 3] CARRY 3 would give the ROCKET shape to Sprites 1, 2, and 3]. The procedure below shows how to talk to all 32 Sprites:

```
TO EXPLODE
TELL :ALL
CARRY :BALL
EACH [SC YN]
SS 55
EACH [SH YN*11] SS 55 WAIT 120 HOME WAIT 120
EXPLODE
END
```

The above program includes some useful primitives for talking to more than one Sprite. YOURNUMBER (YN) which means the number of the Sprite (1-32), and EACH which tells Logo to carry out your instructions for EACH member of a list of Sprites.

When you're finished experimenting with Sprites, type BYE and press ENTER.

BUGBUSTER—If you forget the number or numbers of the Sprites you are talking to, type PR WHO. The numbers of your current Sprites will be printed on your screen.

MAKECHAR (MC) and PUTTILE

MAKECHAR (MC)—All of the letters, numbers and symbols on your keyboard have character numbers assigned to them. Check your **TI Logo** manual for a chart of these. You can use the primitive MC to put any of these characters up on your screen in different ways. You can also make up characters of your own. For example, when you type MC 53, you will see a small grid with the number 5 on it. This is the MC code number for the number 5. MC 90 gives you a Z. MC 32–95 carry existing keyboard characters. To make up characters of your own, use MC with any number from 96–255. Characters are designed in the same way shapes are, using the arrow keys and the FCTN or SHIFT key to fill in the spaces.

To print any of these characters on your screen, type the command PC or PRINTCHAR followed by the code number of the character you want to print. PRINTCHAR can be used with other commands. REPEAT 100 [PC 90] will give you a row of 100 Zs, for example.

PUTTILE—Is a command that allows you to put characters anywhere you want on the screen. Imagine your screen as a grid. Each of the squares in the grid makes up one tile. PUTTILE takes three numbers as inputs. The first is the number of the character itself, the second is the number of the column on the screen, and the third is the number of the row. A chart of columns and rows is printed on the following page.

Character Chart with Character Code Numbers

Group 1

Code Number	Character
32	(space)
33	!
34	“
35	#
36	$
37	%
38	&
39	‘

Group 2

Code Number	Character
40	(
41	)
42	*
43	+
44	’
45	–
46	.
47	/

Group 3

Code Number	Character
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7

Group 4

Code Number	Character
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?

Group 5

Code Number	Character
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G

Group 6

Code Number	Character
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O

Group 7		Group 8	
Code Number	Character	Code Number	Character
80	P	88	X
81	Q	89	Y
82	R	90	Z
83	S	91	[
84	T	92	\
85	U	93	]
86	V	94	^
87	W	95	_

For example: `PUTTILE 90 22 16` puts a Z in column 22, row 16.

Coloring Tiles

The `SETCOLOR (SC)` command is used to set the color of Tiles. For example, to make the Z appear red, type:

```
PC 90
TELL TILE 90
SC 6
```

Look at the chart of characters again. The characters are listed in groups of 8. You can change the color of a whole group of 8 characters by changing the color of just one character in that group. The command for setting the color of a group of characters is `TELL TILE`. For example, `TELL TILE 42 SC 6` sets all 8 characters in group 2 to color #5 or red. You can see your tiles using the `PC` command.

Foreground and Background Tile Colors

A tile has 2 colors, a foreground color and a background color. To set the foreground and background color of a Tile, use the `SC` Command followed by two color code numbers in brackets. The first number sets the foreground color, the second number sets the background color. Your screen is covered with blank Tiles. Use this example to make a tile pattern cover your whole screen:

```
TELL TILE 32
SC [6 13]
```

Your screen background is now purple. All of the blank Tiles have a background color of purple.

Now design a `MC` for the foreground of the Tile. Type `MC 32` to get a blank grid for your design. Watch the screen as your `MC` design shows up in red on the foreground of the blank Tiles across your screen.

Press FCTN and BACK when you finish your design. SC [1 7] restores normal background color. To erase the MC, type MC 32, then press FCTN and CLEAR.

The foreground and background colors of the Turtle can be different, too. To see this effect, type TELLTURTLE and use the SC command followed by foreground and background input numbers in brackets. As the Turtle draws, its lines will be seen in the foreground color, while the background will show up as the second number inside the brackets.

To make the Turtle draw black lines on a clear background again, type SC [1 0].

6

Games and Challenges

On the following pages are some procedures for games, designs, and challenges that we have written just for fun. We have included the complete procedures for each of these programs, as well as some other useful information. The procedures that follow are clearly labeled for the version of Logo they were written on. See if you can adapt them to your system, or start with our ideas and write some new challenges of your own. Happy programming!

GOOD THINGS TO KNOW

PRINT (PR)

Each version of Logo also has a PRINT command. PRINT commands can be used for any number of things. You can have the computer ask you questions, PRINT answers, or include PRINT commands in your graphics programs, like we did in the following program on page 102, ANDY'S TURTLE RACE.

To PRINT one character or one word, use the left quotes sign (“) followed by the letter or word you want to PRINT:

```
PR “D
Result—D
PR “DOG
Result—DOG
```

To have Logo print a list of words or numbers, use brackets [ ]:

```
PR [HELLO COMPUTER PROGRAMMER, WHAT SHALL WE TRY TODAY?]
```

HELLO COMPUTER PROGRAMMER, WHAT SHALL WE TRY TODAY?
PRINT can also be used to do arithmetic. For example:
PR 12+12 Result: 24

TOOT and BEEP

TOOT is a command in **Atari Logo** that can give your program special sound effects. Here's how it works:

TOOT takes four numbers as inputs:

VOICE—TOOT has two voices, 0 and 1. This is the first input number.

FREQUENCY—TOOT measures sound in cycles per second. The choice of input numbers ranges from 14 to above audible sound levels.

VOLUME—This adjusts the volume level of your monitor. The input numbers here range from 0 for very soft, to 15 for very loud.

DURATION—This input number tells TOOT how long to hold the note. It measures time in 60ths of a second, like the WAIT command. The choice of input numbers here ranges from 0 to 255. For example, `TOOT 1 100 5 60` uses voice 1 at a frequency of 100, a volume of 5 and a duration of 1 second.

BEEP is the sound primitive in **TI Logo.** BEEP doesn't use input numbers, but can be used in combination with the WAIT commands for varying effects. If you want to turn off the BEEP in **TI Logo**, use the command NOBEEP.

ANDY'S TURTLE RACE (ATARI LOGO)

This uses collision detection, the PRINT command, and TOOT for an exciting game that turns out differently each time. To see what this looks like, turn to page 61 of the color insert.

```
TO RACE
SET BG 106 CS CT
LINE
TELL [0 1 2 3]
EACH [PU FD 25*WHO RT 90 BK 135 ST]
WAIT 150 PR [THEY'RE OFF!]
EACH [SETSP 15 + RANDOM 35]
WIN
END
```

```
TO LINE
TELL 0 HT PU RT 90 FD 135 LT 90 BK 15 PD
FD 105 PU HOME
END

TO WIN
TELL [0 1 2 3]
WHEN 0 [SETSP 0 PR [TURTLE 0 WINS!] SOUND]
WHEN 4 [SETSP 0 PR [TURTLE 1 WINS!] SOUND]
WHEN 8 [SETSP 0 PR [TURTLE 2 WINS!] SOUND]
WHEN [12 SETSP 0 PR [TURTLE 3 WINS!] SOUND]
END

TO SOUND
TOOT 1 400 15 20
TOOT 1 440 15 10
TOOT 1 400 15 20
END
```

CONNIE’S CARTOON

This procedure animates the Turtle as it draws, and gives you an idea of how it feels about its work!

```
TO CARTOON
FS PD
REPEAT 3 [RT 120 FD 50]
REPEAT 3 [LT 120 FD 50]
FD 50 RT 150 FD 85 PU BK 85 PD
RT 60 FD 85 PU BK 85 LT 30 BK 25
WAIT 300 PU
REPEAT 3 [RT 90 FD 10 BK 10]
REPEAT 9 [RT 90]
REPEAT 4 [LT 90]
RT 180
FD 10 WAIT 30 LT 180 FD 25 PD FD 100 PU FD 25 RT 360 WAIT 30
RT 180 WAIT 120 FD 125 PE FD 10 LT 180 WAIT 60 FD 10 PD FD 100
PU FD 25 WAIT 60 RT 180 PU FD 135 RT 180 WAIT 120 PE FD 20 PU
FD 160
RT 180 WAIT 60
REPEAT 8 [RT 90]
WAIT 60 LT 120 PU FD 250 LT 120 WAIT 60 FD 250 LT 60 WAIT 60
FD 250
LT 120 WAIT 60 FD 250 LT 120 WAIT 120
REPEAT 8 [RT 90]
REPEAT 8 [LT 90]
SETH 0
END
```

PAT'S FISHTANK

FISHTANK (the main procedure) uses only one variable, :SIZE to create a very complex drawing. (This procedure was written for **Apple Logo.**) Type FISHTANK 200 to run this program after typing in all of the procedures.

```
TO FISHTANK :SIZE
SETBG 5 SETPC 0
HT PU SETPOS [-115 -30]
FULLSCREEN PD
PAR :SIZE
RT 30
RECT1 :SIZE
FD :SIZE*.75
PAR :SIZE
RT 180 FD :SIZE/3 LT 60
RECT2 :SIZE
PU HOME PD SETPC 2
WEED :SIZE/10
SETPC 4
SWIM :SIZE
PU HOME
SWIM1 :SIZE
PU HOME
SWIM2 :SIZE
PU HOME SETPC 4
STARFISH :SIZE
SETPC 1
SHELL :SIZE
PU HOME
STARFISH.5 :SIZE
END

TO WEED :SIZE
PU SETPOS [-40 -60] PD
ARCR :SIZE 60
ARCL :SIZE 90
ARCR :SIZE 60
ARCL :SIZE 90
PU SETPOS [-40 -60] PD
SETHEADING 0
ARCL :SIZE 50
ARCR :SIZE 50
ARCL :SIZE 60
ARCR :SIZE 75
PU SETPOS [-40 -60] PD
SETHEADING 0
ARCR :SIZE 45
ARCL :SIZE 85
ARCR :SIZE 70
ARCL :SIZE 65
ARCR :SIZE 40
```

```
PU SETPOS [-40 -60] PD
SETHEADING 0
ARCL :SIZE 70
ARCR :SIZE 80
ARCL :SIZE 30
ARCR :SIZE 60
ARCL :SIZE 50
PU HOME PD
END

TO PAR :SIZE
RT 90 FD :SIZE RT 60
FD :SIZE/3 RT 120 FD :SIZE
RT 60 FD :SIZE/3
END

TO RECT1 :SIZE
REPEAT 2 [FD :SIZE*.75 RT 90 FD :SIZE RT 90]
END

TO RECT2 :SIZE
REPEAT 2 [FD :SIZE RT 90 FD :SIZE*.75 RT 90]
END

TO FISH :SIZE
HT RT 30 ARCR :SIZE/10 90
RT 120 ARCR :SIZE/10 90
END

TO EFISH :SIZE
PU SETPOS [0 0] PD
RT 30
PE FISH :SIZE
PD
END

TO EFISH2 :SIZE
PU SETPOS [30 0] PD
RT 30
PE FISH :SIZE
END

TO SWIM :SIZE
FISH :SIZE PU
EFISH :SIZE
MOVE
PU SETPOS [30 0] PD
RT 30
FISH :SIZE
EFISH2 :SIZE
PU SETPOS [35 0] PD
MOVE
```

```
RT 30 FISH :SIZE
END

TO MOVE
RT 90 PU FD 20 PD LT 90
END

TO SWIM1 :SIZE
PU SETPOS [10 – 10]
FISH :SIZE
MOVE
PU SETPOS [40 – 10] PD
RT 30 FISH :SIZE
PU SETPOS [45 – 10] PD MOVE RT 30 FISH :SIZE
END

TO SWIM2 :SIZE
PU SETPOS [55 – 20] PD
FISH :SIZE
MOVE
PU SETPOS [60 – 25] PD
RT 30 FISH :SIZE
PU SETPOS [65 – 30] PD
MOVE
RT 30 FISH :SIZE
END

TO STARFISH :SIZE
HT PU SETPOS [50 – 85] PD
REPEAT 5 [FD :SIZE/10 LT 72 FD :SIZE/10 RT 144]
PU FD :SIZE/10 RT 140 PD
END

TO SHELL :SIZE
HT PU SETPOS [0 – 50] PD
IF :SIZE/20<0 [STOP]
SHAPE: SIZE
SHELL :SIZE – 40
END

TO SHAPE :SIZE
REPEAT 12 [RT 30 FD :SIZE/20]
END

TO STARFISH.5 :SIZE
HT PU SETPOS [50 – 85]
FD :SIZE/10
PD RT 30
REPEAT 5 [FD :SIZE/30 LT 72 FD :SIZE/28 RT 144]
END
```

BLAST OFF

This procedure puts the Turtle into orbit around Earth. It was written for **Apple Logo**. To see what this looks like, turn to the color insert, page 59.

```
TO BLAST
SKY 250
PLANET1
PLANET2
EARTH 40
PMOON
SET
COUNTDOWN 10
PRINT “LIFTOFF WAIT 60 FULLSCREEN
EXPLODE
ORBIT
END

TO DOTS
DOT LIST RANDOM 280 RANDOM 240
END

TO SKY :DOTS
HT FULLSCREEN
REPEAT :DOTS [DOTS]
END

TO PLANET :STEPS
IF :STEPS < 0 [STOP]
HT PD SETPC [1 + RANDOM 5]
REPEAT 72 [FD :STEPS RT 5]
PLANET :STEPS - .1
END

TO PLANET1
PU SETH 0
SETPOS [- 55 50]
PD PLANET .5
END

TO PLANET2
PU SETH 0
SETPOS [0 0] PD
PD PLANET 1
END

TO EARTH :SIZE
SETPC 5 HT
IF :SIZE < 0 [STOP]
PU SETPOS [- 140 - 117]
SETH 55 PD
REPEAT 6 [FD :SIZE RT 14]
```

```
EARTH :SIZE -.4
END

TO MOON
HT RT 60 SETPC [1 + RANDOM 5]
REPEAT 40 [FD 1 LT 4]
LT 160
REPEAT 30 [FD 1 RT 4]
FD 10 LT 160 FD 10
END

TO PMOON
SETPC 3
PU SETH 0
SETPOS [117 -99] PD
MOON
END

TO SET
PU SETPOS [-40 -80] SETH 0
END

TO COUNTDOWN :X
SPLITSCREEN
IF :X = 0 [STOP]
PRINT :X
WAIT 60
COUNTDOWN :X - 1
END

TO EXPLODE
HT
SETPC 4
PD LT 90
REPEAT 7 [FD 10 BK 10 RT 30]
PU SETH 0 SETPOS [-40 -65] ST
END

TO ORBIT1
SETPOS [126 -112] ST
SETH -45
REPEAT 67 [FD 4 LT 1]
LT 13 REPEAT 3 [FD 4 LT 1] HT
WAIT 240
ORBIT1
END

TO ORBIT
PU SETH -70
REPEAT 53 [FD 2 LT 1.5] FD 3
HT WAIT 240
ORBIT1
END
```

AMAZING MAZES

ATARI LOGO

The following procedures make up a game where you try to move the Turtle through two different mazes. It's not as easy as it sounds, though, because the programs include collision detection. If you hit a line, the Turtle gets bumped out of the maze and has to start over!

MAKE.MAZE1 and MAKE.MAZE2 and the procedures they include teach the Turtle how to draw the two mazes. To see what these look like, turn to page 66 of the color insert.

PLAY1 and PLAY2 set up collision detection and make the screen flash when the Turtle successfully completes each maze. PLAY1 also causes MAZE2 to appear and PLAY2 to begin when the player completes MAZE1.

The procedure TO MOVE sets up four keys on the keyboard with which the player can move the Turtle through the mazes. The four keys are:

F for FORWARD 4
B for BACK 5
R for RIGHT 90
L for LEFT 90

GAME is the "calling procedure." After typing in all of AMAZING MAZES, the player then types GAME and presses the RETURN key to begin.

```
TO MAKE.MAZE1
TELL [1 2 3] HT
FS
TELL 0 ST SETC 7
SETBG 74
MIDDLE 1
SETPN 0
SETPC 0 7
CONSQUARE 60 20
LINES1
END

TO MIDDLE 1
SETPN 1
SETPC 1 15
PU HOME PD
REPEAT 20 [RT 90 FD 1 LT 90 FD 20 BK 20]
PU HOME LT 90 FD 20 RT 90 BK 20 PD
END

TO SQUARE :SIZE
REPEAT 4 [FD :SIZE RT 90]
END
```

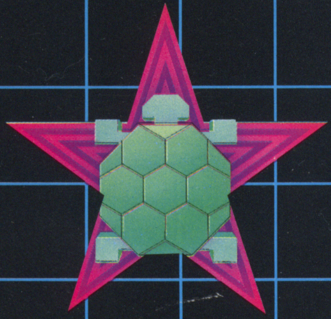
```
TO CONSQUARE :SIZE :DIS
IF :SIZE>220 [STOP]
SQUARE :SIZE
PU LT 90 FD :DIS RT 90 BK :DIS PD
CONSQUARE :SIZE + :DIS*2 :DIS
END

TO LINES1
PU HOME
RT 90 FD 40 LT 90
PE BK 20 RT 90
FD 20 LT 90 FD 5 PE FD 20
LT 90 BK 20 RT 90 PD BK 30 LT 90
BK 20 PU BK 20 RT 90
FD 5 PE BK 20
PD BK 60 LT 90
FD 110 RT 90 FD 20 LT 90 FD 90 RT 90
PE FD 20 RT 90
FD 20 LT 90 PD FD 140 PE BK 20 RT 90
FD 20 RT 90 PE FD 20
PD FD 55 LT 90 FD 20 BK 20
RT 90 FD 25 LT 90 FD 20
PE FD 20 PD FD 5 RT 90 FD 20
PU BK 120 PD BK 20
PU RT 90 FD 105 LT 90 PE FD 20
PU SETPOS [-130  -20] SETH 0
END

TO MIDDLE2 :SIZE
IF :SIZE<0 [STOP]
REPEAT 3 [FD :SIZE RT 120]
MIDDLE2 :SIZE - 1
END

TO MAKE.MAZE2
TELL 0 ST SETC 75
FS SETBG 34
PD SETPN 1
SETPC 1 15
RT 30 MIDDLE2 20
PU HOME PD
SETPN 0 SETPC 0 7
CONSQUARE 40 20
PU HOME
LINES2
END

TO LINES2
LT 90 PE BK 20
PU FD 60 PD BK 20
RT 90 FD 100 RT 90 FD 100 PE FD 20
PD FD 20 PU BK 200 LT 90 PE BK 20
```

```
PU FD 20 RT 90 FD 100
LT 90 PD FD 20 PU BK 140 LT 90 FD 80
RT 90 PE FD 20
LT 90 PU BK 140 RT 90 FD 40 PE FD 20
PU BK 60 LT 90 PD FD 20 PU LT 90 FD 40
RT 90 FD 60 PE FD 20
RT 90 PU FD 100
RT 90 FD 10 PE FD 20
RT 90 PU FD 140 LT 90
FD 60 PE FD 20 PU FD 30 LT 90 FD 100
PE FD 20
PU SETPOS [ - 128  - 60]
END

TO FLASH1
REPEAT 10 [SETBG 34 WAIT 15 SETBG 74 WAIT 15 SETBG 73
WAIT 15]
SETBG 74
REPEAT 10 [SETC 45 WAIT 15 SETC 22 WAIT 15]
END

TO PLAY1
WHEN 0 [SETPOS [ - 130  - 20]]
WHEN 1 [FLASH1 WAIT 300 CS MAKE.MAZE2 PLAY2]
END

TO FLASH2
REPEAT 10 [SETBG 74 WAIT 15 SETBG 2 WAIT 15 SETBG 99
WAIT 15]
SETBG 34 REPEAT 10 [SETC 55 WAIT 15 SETC 22 WAIT 15]
SETC 75
END

TO PLAY2
WHEN 0 [SETPOS [ - 128  - 60]]
WHEN 1 [FLASH2]
END

TO MOVE
MAKE "KEY RC
IF :KEY = "R [RT 90]
IF :KEY = "L [LT 90]
IF :KEY = "F [FD 4]
IF :KEY = "B [BK 5]
MOVE
END

TO GAME
TELL [1 2 3] HT
MAKE.MAZE1
PLAY1
MOVE
END
```

Two new primitives, MAKE and RC, are introduced in the recursive procedure TO MOVE.

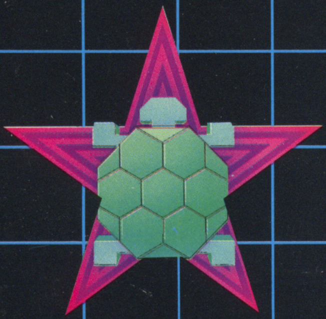
MAKE is another way of creating variables in Logo. Here, MAKE creates KEY as the variable. KEY can have the values "R (RT 90), "L (LT 90), "F (FD 4), or "B (BK 5).

RC is the abbreviation for READCHARACTER. Using RC, the keyboard character pressed by the player is looked at (or READ) by Logo. In TO MOVE, IF R, L, F, or B is pressed, the Turtle moves or turns according to the commands in brackets next to each of these characters. If any other character is pressed, nothing happens.

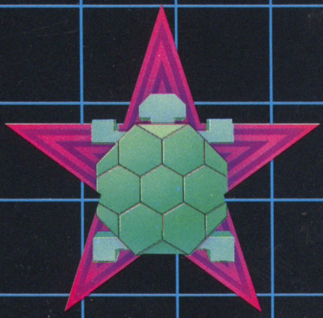


TURTLE TALK

Did you know that there's a turtle living inside of your computer? It's a new kind of turtle with a language all its own called LOGO, and it's the first computer language you can command to draw! Teach the turtle to talk, and you will be able to draw a star in a turquoise sky, a tarantula in a web, an airplane flying over a golden pyramid. TELL TURTLE to PENDOWN and start to program designs of your own—people, circles, arcs, hexagons!



LOGO FUN is for everyone. Anyone who can read can learn LOGO, and anyone can talk turtle, it's so easy. With programs illustrated in black-and-white and color, you'll learn to create pictures, games, and projects on your home computer. LOGO's an illustrated language that's easy and fun. Take a Total Turtle trip with a few Turtle Steps and you'll find out more about programming and your world than you ever knew before. The sky's the limit and the turtle's your new friend. So start talking turtle!



SCHOLASTIC INC.

33243

0 78073 00595

ISBN 0-590-33243-0

Cover Painting: Stanislaw Fernandes